

YOUR  **PC**

SEPTEMBER 1989 £1.30

COMMODORE

BATMAN

Waiting for a dark knight

GREENPEACE

Fight for the Foliage

GAMES

- Thunderbirds
- Rockstar ate my Hamster
- Escape from Drill
- Phobia



LICENCE TO KILL

007 – On his own and out for revenge

ISSN 0269-8277



9 570259 827007



CONTENTS

007TM LICENCE TO KILL



- **Loose Wolf** 8
The game of the gamebooks
- **James Bond** 17
Domark is back
- **Batman** 22
At 50 is Batty past it?
- **Greenpeace** 28
We've got smoke. We've got radiation...
- **News** 4
- **Contributions** 12
So kid, you'd like to write for us?
- **Four Ground** 14
Plus 4 winners stop here
- **128 Corner** 20
Are you a troubled 128 user?
- **PC Corner** 26
The regular series for the Commodore PC
- **Next Month** 45
- **Big Wobbly Comps** 37
We look at Thunderbirds games
- **Software for Sale** 70
You need to type!
- **Extending Basic** 76
Part the sixth
- **Back Page** 82
Hot juicy gossip and The Noddies

- **Thunderbirds** 62
1... 2... 3... 4... 5...
- **Gilbert** 64
Get smelly!
- **Super Scramble** 65
Are that engines?
- **Red Storm rising** 66
Nuclear conflict, comrades
- **Storm Lord** 67
Oh what a fury!
- **Rockstar At My Hammer** 68
And it were right tasty
- **Ultimate Twisting** 69
Those days of Role Playing
- **Terminator** 72
Time travelling trauma
- **Fire King** 73
What, more role playing?
- **Phobia** 74
I can't stand it any more, I've got Google phobia
- **Journey to the Centre of the Earth** 75
Cos, it's a bit hot down here, isn't it?

- **Dogs** 29
Get your hands round your tool
- **Boxes and Squares** 42
Tutorial into mathematical formulae
- **Reasoning on the 128** 46
Continuation of the expert system
- **Print FX** 48
Correction for our previous errors
- **DOS 4.1** 54
Enhanced DOS commands for disk users

FEATURES

REGULARS

PROGRAMS

GAMES

Editorial

If you flick through the pages of this month's *Your Commodore* you may find a few subtle (and a couple of blatantly obvious) changes. This is because the editorial team, upon my arrival in the field, has decided to offer a brighter future for *VC*, one that we can all enjoy together.

My first major decision was to revolve this issue around *Burnin'*, and I hope you all agree that this was a wise move. Apparently the movie grossed over \$40,000,000 in its first weekend, and seeing as it cost \$18m less to make it has more than doubled any fears from Warner Brothers that it was to flop, and this is before it reaches British shores. Further on in this issue you can read about the reasons the film was undertaken, and a preview of the game.

Other features of note are on *Lynx* to *SEI* - the new Bond film, *Grease* and *Love Walk*, so make sure you've got your asbestos gloves on, they are HOT!

Although there are many entertainment based articles we haven't left out the usual listings, or the machine based columns. The programs can now be found in a section of with their own, so you can keep them separate from the rest.

I hope you enjoy this issue as much as we've enjoyed putting it together and that leaves me to say, welcome to the beginning...

Ed Henderson

Deputy Editor: Richard 'Boy Wonder' Henderson

Technical Editor: Paul 'Riddler' Eves

Group Editor: Stuart 'Penguin' Cooke

Advertisement Manager: Paul 'Rayzicle' Kavanagh

Sales Executive: Marie 'Circusman' Wade

Ad-Copy Control: Karen 'Maggie' Barclay

Artists: Alan 'Bashy' Saville, Alan 'Kolar' Lovell

Designer: Mark 'Dark Knight' Newton

Origination: Blissy

Printers: Chase Web

The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights reserved by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Ltd and any reproduction

requires the prior written consent of the Company, 1989.

Distribution: BM Distributors, 4 Lifford Court Road, London SW15 2PG. Printed by Chase Web, Plymouth.

Opinions expressed in reviews are the opinions of the reviewers and are not necessarily those of the magazine. While every effort is made to thoroughly check programs published for errors we cannot be held responsible for any errors that do occur.

Your Commodore incorporating *VC* is a monthly magazine appearing on the first Friday of each month. Argus Specialist Publications Limited, Editorial and Advertisement office, Four Commodore, Argus House, Boundary Way, Blandford Dorset DT11 7SD. Telephone: (0442) 24001. Subscription rates upon application to Your Commodore Subscription Department, Inform Ltd, 5 River Park Road, Bournemouth, Bournemouth BH1 1BL. Telephone: (0442) 311175. U.S.A. Subscription Agent: Wise Owl Worldwide Publications, 4114 West 238th Street, Torrance CA 90505 U.S.A.

Data Statements

The Bug Bites Back

Now that *The Bug*, the incredibly successful computer journal, has officially folded, editors Jason Lewis and Jeffrey Davenport are to sell their entire software review library to raise funds for charity.

The collection includes five years of software (business and otherwise) and a majority of the profits are to go to a consortium of separate charities and voluntary organisations.

If you wish for more information, the current list, etc, write to 28 Lonsdale Avenue, Mowbray Hill, London N10 3BU. Please include an SAE.

EA's Horror Story.

Electronic Arts is to bring the horrific world of H.P. Lovecraft's *Cthulhu* myths onto Amiga and PC screens in September. The *Horror of the Shadow* is a role playing game set in the roiling 20s, in which the player must use investigative research in order to unravel the secrets of arcane lore and discover many vicious horrors lurking beneath seemingly normal life.

The game system that was developed by board game designers and fiction writers, *Cthulhu* Games, allows the player to create a character by selecting sex, nationality, one of six professions, and proficiency in over 50 skills. These will directly affect the outcome and his success in the game.

The *Horror of the Shadow* will appear first on the PC in early September and later on the Amiga. Both versions will cost £24.99.



10

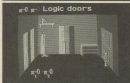
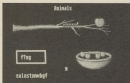
Trendy Lefties in Fun School Controversy

Fun School 2, Duxbury's excellent teaching software, has been rejected for use in schools by 'Trendy' authorities, who believe that it emphasises the 3Rs too much.

The 3Rs (Reading, Writing, and 'Rithmetic') are currently being weeded out of the national curriculum by such left-wing educationists for the 'old fashion' methods of teaching used.

Peter Davidson, one of the co-authors of the software, states "What they don't like about it - apart from it being based on the 3Rs - is that it involves an element of competition, with a risk coming up if the question is answered correctly".

Shelly Gibson, a teacher who has been using **Fun School 2** in her class at Poplar Street Primary School, Audenshaw, has written to the Minister of Education, Kenneth Baker, to urge him to explore matters further.



PC Formula One.

Electronic Arts is to release a PC version of its best selling **Amiga** racing game, **Ferrari Formula One**. Planned for an early August date, the game aims to bring together the excitement of steering in a **Formula One** championship against Alain Prost, Nigel Mansell and Ayrtton Senna and the strategy of managing a team.

The game will feature all 16 tracks from the 1984 season including Monaco, Brazil's Hung and Monza

plus two additional tracks so that the player can compete in the current season. Each track is modelled accurately down to its length, the background and the weather conditions the drivers will face.

The races themselves aim to put you in the middle of the action as you attempt to defeat your Ferrari to victory. **Ferrari Formula One** will cost £29.99. Available on both 3.5 and 5.25 inch will support both CGA and EGA graphics.

Socket to Them

Computers, monitors and TVs are all vulnerable to mains interference. This can damage hardware, corrupt data, and cause games to crash, not ideal when the high score is about to be broken. The most common type of interference is a mains 'spike', provided by home equipment such as washing machines, power tools, and microwave ovens. The same applies with your neighbour's electrical equipment, or even the Electricity company having problems maintaining a stable voltage.

One solution to such a problem comes from Apollo Electronic Products Ltd, who have introduced a new Apollo Spike Suppressor which, for the price of £12.50



for a plug unit, can protect your computer system. These British made devices are simple to use, the most you have to do is replace the mains plug.

The range also includes a 3-way adapter for £16 and a 4-gang strip for £35 (prices include P&P and VAT). All the units are capable of absorbing a spike of 4500 Amps/215 Joules.

For more information contact Apollo Electronic Products Ltd, on (04858) 8136.

Game, Set and Mat

Your Commodore are proud to present mouse mats and dust covers for the A500 and C128. The covers are especially designed to protect your computer from dust, grime, coffee, tea, etc, ensuring your circuits do not get clogged by spillages and internal acid rain.

Made exclusively for your favourite computer mag, they are stylish and robust, and come complete with

their own nifty logo.

The mouse mats too are emblazoned with the 'Your Commodore' logo, and they help to allow your mouse grip on any surface.

At the price of £4.45 for the covers and £4.65 for the Mouse Mat, they represent terrific value for money.

They are available from: **Readers Services, Argus House, Boundary Way, Hemel Hempstead HP1 1ST.**



It's A-Maze-ing

Brown's latest game comes our way in late August. Titled *Mazemonster*, it features a character called Flippo who must travel through the maze of Tabular Rasa turning tiles as he goes - hence his name Flippo.

Gliding along kinder his quest, but there are lions to supply him with extra power. Once granted, the maze will fade away to leave Flippo with yet another to

tackle.

The Amiga version will have 30 levels to complete, while the C64 will only feature 12-16. The price of the C64 version will be (on cassette) £9.99.



Anyone for Tennis

Miramax, under their Imageworks label, has secured the rights for the Sega coin-op, *Passing Shot*. Programmed by Tapes - the team behind *Thunderbirds* and *Masterforce*, among others - it captures all the options of the arcade

machine and will be available on both the C64 and the Amiga.

Passing Shot features doubles and singles championships, taking place on clay and grass courts, in an effort to win the elusive Grand Slam. It offers an impressive variety of gameplay options with the player controlling both the posi-

tioning and type of shot (lob, spin, etc.). The court scrolls to allow you to follow the action and there are two perspectives to encounter during the game - overhead for gameplay, HD for service.

Ball boys, the net judge and the line-man are all there and we can expect to see it in late Autumn.



Oil See You in the Morning

Rainbow Arts, the company responsible for the scandalous *Detour/Katalka*, has undertaken a different kind of project.

Released on the Redline Software label, *Oil Imperium* deals with the cut and throat world of the oil business. You can become J.R.

as you try to put your competitors out of business. What is even better is that these enemies can be your friends, or the computer.

Bullying, selling, trading, spying and sabotaging are all involved in getting to the top, so only the devious minds may apply.

Oil Imperium will be available in September on the C64 and Amiga!



Only the Lonely

The Lone Wolf bandwagon continues to roll along the road of success. Rik Henderson hitches a ride and prepares for a one-way trip to fantasy land.



plus the downfall of the Kai, the warrior elite of Summerland (as it quotes on the back). Although it contains many good fantastical ideas, there is always a feeling of déjàvu connected.

The Dark Door Opens is not much better in theory, but perhaps a more enjoyable read. Now that the Kai have been eliminated, Lone Wolf is born and the plot follows his quest to reach King Ulmar and warn him of impending doom.

Analogue Software has now owned the effect of Lone Wolf, and thus released a new computer game based on his exploits.

Entitled simply *The Mirror of Death*, the game places Lone Wolf in an arcade-adventure. Although one could be forgiven for thinking that it has more than a smidgen of hack-and-slash action more usual in a beat-'em-up.

The plot follows Lone Wolf as he tries to recover a stolen gem, once belonging to his mentor, the Kai. This is no easy task as the tower in which it is imprisoned is full of fire-spitting gargoyles, demons and countless other fatal dangers. Once all these have been negotiated, Wolf enters face-to-face with the Mirror of Death itself, which is not termed such for nothing. It has many properties, but its most famous, and arguably its most lethal, is the fact that it can produce carbon-copies of its enemies. Not just any old reproduction mind you, but one that has evil intentions. It will take more than a quick slash to vanquish this fellow.

Lone Wolf - The Mirror of Death boasts great graphics and good action. The only reservation that I have is that the play may not be varied enough, thus giving it a short life.

It is available for the PC, Amiga, and C64 at varying prices.

Other news comes out way that *Mirror of Death* will not be the last Lone Wolf excursion into the world of home computers. Analogue has many more adventures planned for

The *Lone Wolf* Legends are a new series of books by Jan Dever, who hopes to emulate the success of previous efforts. Co-written with John Grant - whose credits to date include *Demons*, *Evilbeasts* (co-written with David Langford) and, strange as it may seem,

the *Encyclopedia of Walt Disney's Animated Characters* - they are not short of technical brilliance, although one wonders whether the fantasy orientated plot is a little too clichéd.

Belgore of the Kai is the first in the indistinguishable number of books. It deals with the origin of Lone Wolf

1990).

Last, but certainly not least, in the new Lone Wolf releases is the inter-



active telephone line, written by Joe Dever. This is surely in direct competition with Steve Jackson's *Pit*.

The company involved, *Broadway*, has invented a new label under which they hope to release many games of this kind, *Forces of Deceit* being just the first. *Phonquest* (T&H) games will all use the same system of operation, although at this stage it is hard to determine whether they'll just be sticking with Lone Wolf, or if they may branch into other areas of interactive 'quests'.

Unlike *Pit*, which uses a time method to determine actions, *FOD* breaks are grounded with voice-interactive technology. This allows the caller to make 'real-time' decisions during play, and increases the tension created by the stunning sound-effects.

The way in which this works is simple. When at a point in the game where a positive decision is to be made, you will be given a number of options (like the Lone Wolf gamebooks), each separate option carries a different number which must be guessed after a 'clue', thus allowing the action to be performed. Aggressive moves are similar, although they must be guessed when in battle, and a sticky and is almost a certainty.

The sound-track was recorded at a top London studio using a very able team of professional actors, with many other effects added to make the game an all-round fantasy experience. Separate sections of the adventure are narrated by different voices (male and female) and they all do a great job in the atmosphere stakes.

One right I have of any game like

this is the prices which must be charged. CD peak rate is 23p per minute, whilst peak is 39p per minute, costing a mammoth £15 for a hour's play (off peak), which will add up to £145 per week, if it is played for an hour a day, quite a sum of money.

This is most worrying for the younger children who become addicted to such things (hence the reason for *Charlie's* demise). My advice to anybody worried about the costs, buy one of these gamebooks, at around £2.99, and get a friend around to read it out to you, it's not that much different and certainly a lot cheaper.

I'm not here to condemn the line, I think it's rather good fun, and if played in moderation it's a very worthwhile venture.

All the Lone Wolf products have something going for them, and the fact is that they'll be popular for a long while yet, thanks to *Joe Dever's* persistence, and *Broadway* looks money, for when both are flying legends can be born.

Lone Wolf Bibliography

Lone Wolf Gamebooks

1. *Flight From The Dark* - Joe Dever & Gary Chalk
2. *Fire on the Water* - Joe Dever & Gary Chalk
3. *The Carrion of Kalis* - Joe Dever & Gary Chalk
4. *The Chasm of Doom* - Joe Dever & Gary Chalk
5. *Shadow on the Sand* - Joe Dever & Gary Chalk
6. *The Kingdoms of Tantor* - Joe Dever & Gary Chalk
7. *Castle Death* - Joe Dever & Gary Chalk
8. *The Jungle of Horrors* - Joe Dever & Gary Chalk
9. *The Cauldron of Fear* - Joe Dever
10. *The Dungeons of Torgar* - Joe Dever
11. *The Prisoners of Time* - Joe Dever
12. *The Masters of Darkness* - Joe Dever

The World of Lone Wolf Gamebooks

1. *Grognax the Wizard* - Joe Dever & Ian Page
2. *The Forbidden City* - Joe Dever & Ian Page
3. *Beyond the Nightmare Gate* - Joe Dever & Ian Page
4. *War of the Wizards* - Joe Dever & Ian Page

Legends of Lone Wolf

1. *Edipian of the Kai* - Joe Dever & John Grant
2. *The Dark Desert Opens* - Joe Dever & John Grant

Others

- The Magusworld Companion - Joe Dever & Gary Chalk
- The Lone Wolf Poster Painting Book - Gary Chalk



DATTEL ELECTRONICS

THE ULTIMATE GRAPHICS PACKAGE...



MOUSE
ALSO DEC COMPTABLE

SPECIAL OFFER PACK



INCLUDES FREE MOUSE MAT/HOLDER WORTH £12.95



COMMODORE 1351 MOUSE

- 1351 Mouse is a high resolution two button mouse featuring optical counting, teflon guides, microswitches, rubber contact ball and high quality interface.
- When combined with DCP Advanced Art Studio this graphics package is quite simply the best system available. Its features are unparalleled...
- Works on image - works to expand it, moves it, rotates it, crops it, zooms it, etc. etc.
- Spans preference to shades, make colour tables - colours are completely chosen.
- Mouse is to add detail to your work.
- Preference to the mouse is to add detail to your work.
- Mouse operation, plus joystick and keyboard control.
- 16 pins, 16 colours, 16 lines - no trouble, mouse can make reports graphics ready.
- Full set and your facilities plus excellent printer support.
- Full set, full screen, full screen, colour, with no features (16) and professional manual make an absolute mouse (the best graphics package) for the Commodore system (see Page 10) (see).

COMPLETE WITH

DCP ADVANCED ART STUDIO

ONLY £29.99

TOTAL PACKAGE WITH MOUSE 1351 MOUSE (WORTH £12.95) HOLDER AND DCP ART STUDIO



VIC 20 RAMPACK

- Master (30) to memory expansion and other (10) to memory expansion.
- Two models available - 128 or 256.
- Both models are available in the various memory expansion (10, 128, 256, etc).
- Master (30) program expands memory (10) to (10) to memory.

ONLY £14.99 (see 128)

ONLY £19.99 (see 256)

HOW YOU CAN INCREASE THE MEMORY OF YOUR VIC 20 EASILY



C16 RAMPACK

- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.

ONLY £14.99

EPROM ERASER

- This handy device (10) to memory expansion (10) to memory expansion.
- Works to (10) to memory expansion (10) to memory expansion.
- Works to (10) to memory expansion (10) to memory expansion.

ONLY £29.99

ENTER NUMBER 12 IN CARD ONLY A CARD ONLY

256K SUPEROM EXPANDER



- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.

ONLY £29.99

EPROMMER G4



- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.

- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.
- 16 to 128K memory expansion (10) to (10) to memory.

ONLY £29.99

COMPLETE

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0203 744787 (after 9.00am - 5.00pm)	Send cheque/P.O. order payable to "Datatel Electronics"	0203 744292

PRICES AND INFORMATION CURRENT AT TIME OF PRESSING AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLER REQUEST: Please quote guide to complete your order.

DATTEL ELECTRONICS

DATTEL ELECTRONICS LTD., FORTON INDUSTRIAL ESTATE, OCEAN ROAD, FORTON, STURGEON-TRENT, ENGLAND

SALES ONLY
0203 744787

TECHNICAL ONLY
0203 744292

Contributions



*So you have written a program? You want it published?
Follow these guidelines when you send it in to us!*

Your Commodore is always on the lookout for new programs, hints, tips, articles and even regular series. In fact, if you have something that you think could be of use to other Commodore owners, we want to hear about it.

Below you will find a list of guidelines that will help us to deal with any hint you send in. We don't expect everybody to be building 'William Shakespeares', although if you do follow these simple rules it will make our job a lot easier. (And speedier).

We are constantly striving to make our magazines more professional in their approach. Therefore, it is to this end that we are making some alterations to how programs should be submitted with a view to publication.

In future, whenever you submit programs or articles for possible publication, we ask you to ensure it is submitted according to the guidelines set out below. Please note:-

Any program NOT submitted along these guidelines will automatically be rejected.

Submission Guidelines

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double spaced, ie there should be a blank line between each line of text. There should be a margin of at least 10 characters around the text.

3) The very first page should consist of the following:

Name of the article/program
Machine that it is for
Any extras needed - drive/ printer, etc.
Your name, address and telephone number.

4) The top of every page should have the following information on it:
Program title
Your name
Page number

For example:

Rasters A, B, G, G, Page 1

5) Do not make additional marks on your text, especially underlining.

6) On the bottom of each page put the word MORE if there are more pages or END if it is the last one.

7) If possible enclose a listing of your program. If written in machine code, then a listing of the source code would be appreciated.

8) Programs should be on either tape or disk with TWO copies included. If problems appear then we have more chance of successful loads. If the program is less than 10 lines it can be included in the text.

9) If your article needs any artwork then supply clear examples of what is needed. We don't expect you to be a first class artist but we do need to see what is required.

An Introduction To Plus/4 Machine Language

Mark Everingham reveals the inner workings of his Plus/4

In future installments of *FourGround*, we shall be printing many articles and programs which need a basic understanding of Machine Language programming. For this reason, over the next few months *FourGround* presents a brief beginners course to machine language. It is obviously not possible to present a full discussion of the subject within the limited space of the magazine, but the course aims to provide a foundation on which to build more advanced knowledge.

For most people, the phrase "machine-code" has horrible images attached to it - strings of binary, incomprehensible listings and degree level mathematics. However, the fears the majority have about machine-code are totally unfounded - the language can be easy and extremely rewarding to use. If you have tried to learn machine-code before and given up in frustration, don't run off now, most books on the subject tend to be extremely badly written, full of jargon and irrelevant information. This short course has been written to be compact and easy to understand while retaining all the essential information. For those who can already program in machine language, it should serve as a useful reference work.

What Is Machine-Code?

"Machine-code", "Machine Language" or "Assembly Language" is a computer language just like any other - Basic, PASCAL or C. Machine Language is the native language of the Plus/4 and is that the computer at its most basic level cannot understand anything else. Basic may seem like the computers natural language, but is really nothing more than a program written in machine language. At the heart of your Plus/4 is a silicon chip called the 7800 Central Processing Unit (CPU). When you write programs in machine language, you are sending instructions directly to this chip. When you write in Basic, your Basic programs have to be converted into machine-code as they are run, because the 7800 cannot understand them in their Basic form.

So why program in machine language? The reason most people use Basic is because it is so easy to learn. For example, to draw a circle in Basic takes just one command. In machine language, hundreds of individual commands are needed.

This might make Basic seem far more attractive, but the problem with Basic is that it runs incredibly slowly. Machine language runs commands at up to a million instructions per second, whereas with Basic you're lucky if you achieve 60 per second. Another advantage of machine language is that because you're working at such close proximity to the actual hardware of the machine, you have the ultimate in power over every individual function that the Plus/4 hardware can perform. Once you have begun programming in machine language, Basic will, I can assure you, seem horribly inefficient.

Number Bases - Decimal, Binary, and the Dreaded Hex!

Computers are number machines - Your Plus/4 computer knows nothing about anything but numbers. Everything it does concerns numbers, even the commands it executes are stored as numbers. We human-beings tend to use the number base called decimal or denary, simply because we happen to have ten fingers on our hands, but computers don't like decimal, so instead they use "Hexadecimal" or "Binary".

The concept of number bases is really simple. Think of a decimal number, say 12345. The key to understanding the magnitude of the number is column headings. Look at the number 12345 written under column headings below.

Heading:	10000	1000	100	10	1
Value:	1	2	3	4	5

In order to work out what number is written down, we just add up the products of the values and headings: $1 \times 10000 + 2 \times 1000 + 3 \times 100 + 4 \times 10 + 5$. Each column heading is a power of 10, so the first one is $1, 10^1, 10^2$, the second one 10×10 , the third $10 \times 10 \times 10$ and so on. The principle for hex and binary is identical, except that for hex, the base is 16 and for binary it is 2. This means that the column headings are 1, 16, 16×16 etc. for hex, and 1, 2, 2×2 etc. for binary. To convert a number to decimal, just repeat the process above using the new column headings. For binary, this is easy, as each column can only hold a number 0 or 1. The number 176 is shown here:

Heading:	128	64	32	16	8	4	2	1
Value:	1	0	1	0	1	0	1	0

The final value is calculated as $16(12) + 16(3) + 16(1) + 16(2)$. The same method is applied for hexadecimal, except that if the base is 16, each column needs to hold a number between 0 and 15, whereas in decimal only between 0 and 9. This means we need 4 more digits. For these we use the first six letters of the alphabet, A=10, B=11, C=12, D=13, E=14, F=15. The number 4660 is shown below:

Heading	4096	256	16	1
Value	1	3	1	4

Of course, if in a program we used some numbers which were binary, some decimal and some hex, we couldn't tell the difference - the number 101 could be 101 decimal, 5 binary or 257 hex. The answer is that binary numbers are preceded by a "B" (percent) sign, hex by a "H" (String/ Dollar) sign, and decimal numbers have no prefix. This avoids all possibility of confusion. For most of the time, however, you will find you can hexadecimal, so the problem need not arise.

So why do we use hex and binary instead of decimal? Computers are electronic devices, made up of banks of transistor switches. A switch can only be On or Off, and these settings can be effectively represented by 1 and 0. These 1's or 0's are called bits - Binary Digits referred to by their common headings 0-7 (right-left), and collections of eight of them (255 decimal) are called bytes. Hexadecimal is used because it is much shorter to write down, while retaining a similarity to binary. The number 255 in decimal needs three digits, in binary, 8, but in hex it needs only two.

The Importance Of Numbers

So far we have only talked about numbers for numbers sake, and this is the criminal error which most machine-code writers make. In machine language, nothing but numbers are used, but the important thing to remember is that a number can mean anything. In real life, numbers can be house numbers, telephone numbers, prices, times, or a host of other things. Machine language is identical, numbers can be colours, counts, commands, speeds or anything you wish. All of the commands discussed in the section "The Instruction Set" perform operations on numbers. If you can remember that these numbers can be of any significance, machine-code programming will be elementary!

The Memory Machine

Your Plus/4 is described as a 64K computer. This means that it has enough memory chips to store 65535 byte long binary numbers. You might expect 64x1000, but 65535 is equivalent to 64x1024, 1024 being near enough to 1000 which is a nice binary number. Because of the binary system, each of these 65535 "locations" can hold a number in the range 0-255, and each location is referred to as "address". This is simply a number between 0 and 65535 which points to a byte of memory. In addition, your Plus/4 has two types of memory: ROM (Read Only Memory) and RAM (Random Access Memory). The ROM holds the machine-code program from BASIC. It lives at address 38000-3FFFF, and cannot be changed. The address space 38000-3FFFF is the RAM, although the extra 32K is found at 38000-3FFFF, under the ROM. For the moment we will only consider the lower 32K.

Each byte of RAM can be put to a different use. To find out which byte of RAM is used for what, you should

consult a manual such as ANCO's "C16-PLUS/4 REFERENCE BOOK" which has a full list. Many locations are used for things like the screen, Basic calculations or Basic programs. When you write a program in Basic, the Basic program in ROM finds a place in RAM to store the program and puts it there. In machine language, you have to tell the Plus/4 where to put the program, and you must take care not to put it in a place which is used by another part of the computer!

The Tedmon Machine-Code Monitor

On almost all computers, writing a program in machine-code means that you have to go out and buy a piece of software called an Assembler, lucky for you, the Plus/4 has a simple one built in, called TEDMON.

Machine-code programs are written like Basic as a series of text commands, such as "TXA", "LDA \$8000,X" and "RRR". Each command is represented by a three-letter mnemonic, or abbreviation. To actually run these commands, they must be converted into a series of numbers and stored in RAM. This process is called Assembling.

The TEDMON program on ROM in your Plus/4 will carry out the simple assembly process for you to save you doing all the work. To assemble a program, type "MONITOR" from Basic. In order to assemble a program, you have to tell Basic where to put the program. Consult a memory map if you have one to find a suitable place, but for most of the time, a good place to put programs is at \$8000. This location in memory leaves you with 8K free for Basic even with a graphics screen in operation, leaving 8K for your machine-code program - an amount you are not likely to use up for a long, long time! The TEDMON assemble command takes the following form:

A \$aaa,xxx opened

The "A" is the actual assemble command, and can be replaced by a "." (period) if you prefer. "\$aaa" is the address at which you wish the line of machine-code to be stored. It should always be expressed in hex and provided by a dollar symbol. "xxx" is the three-character mnemonic of the command you wish to be assembled, and "opened" is the opened or parameter the command takes. When you enter a line and push (Return), the line is stored in memory, and a new address is printed, allowing you to enter the next line. All numbers entered into TEDMON should be hexadecimal, and preceded by a dollar symbol.

When you have finished entering a program, delete the address on the screen. You can exit to Basic by pushing "X" and (Return) for exit, or you can run your program by typing "O" plus the first start address of your program. The dollar symbol should not be entered.

Once your program is in memory, you can look at it or edit it by typing "D" plus the address at which you want to start looking. A series of lines will be displayed on the screen. This is called Disassembly, and the lines on the screen can then be edited and re-entered using the normal Screen Editor functions.

We have not covered here every function of TEDMON, but only the most important ones. Full coverage of the TEDMON monitor can be found in your Plus/4 manual.

7501 Processor Registers

As I have said, all of machine-code's commands operate

on numbers. These numbers can be held in three places — in a memory location, in the memory directly after the machine-code command, or in one of the 7501 registers, which are like fast, internal bytes of memory.

The Program Counter

The program counter (PC) is the only 7501 16-bit register. In it is stored the address where the next command to be executed can be found. When an instruction is executed, the PC is automatically incremented so that it points to the address of the next instruction. When you direct control to another location using a JMP, JSR or branch instruction (see "The 7501 Instruction Set"), the address in the PC is changed accordingly, so you don't have to worry about it.

The Status Register

The status register (SR) is an 8-bit (byte) register which is used to indicate the status of various options of the 7501 CPU. In fact, only seven of the eight bits are actually used, and the functions of these bits are discussed below. Each bit is known as a "flag".

Bit 7 - The N (Negative) Flag. The negative or N bit is set to 1 when the last numeric result had bit 7 set, and is/it's to 0 when the number had bit 7 cleared. Effectively, numbers over 127 set the N bit, and those below clear it. If bit 7 is used to show the sign of the number, the N bit reflects this sign accordingly.

Bit 6 - The V (Overflow) Flag. The Overflow (V) bit is set to 1 when the last operation resulted in what is known as a two's complement overflow. This is only of use in signed arithmetic, where bit 7 represents the sign (175, 804). Overflow is set when the status of bit seven has changed incorrectly. In such cases as two numbers yielding a result greater than will fit in one signed byte.

Bit 4 - The B (Break) Flag. The Break bit is set to 1 when the last operation was a "BKE", or is cleared if the operation was anything else.

Bit 3 - The D (Decimal) Flag. Setting the decimal bit enters the decimal mode of the 7501 processor. Clearing it reverts to binary arithmetic. For information on the decimal mode, see the SED instruction.

Bit 1 - The I (Interrupt) Flag. Setting the interrupt (I) bit to one causes interrupts to be disabled, or clearing enables interrupts. The function only works with IRQ (Interrupt Request) interrupts, and is covered fully in the section on interrupts.

Bit 1 - The Z (Zero) Flag. The zero bit is set to 1 when the result of the last operation was zero, for instance when subtracting a number from itself. If the result was any number but zero, the zero bit is cleared.

Bit 0 - The C (Carry) Flag. The carry bit is set to 1 when the last addition gave a result which could not be expressed in 8 bits. When subtracting, it is cleared if a borrow out

of eight bits was required. The bit is also used by the shift and rotate instructions, where the last bit is shifted into the carry flag.

The Accumulator (A)

The accumulator, or A register is the most important of the 7501 registers. It is an 8-bit register which is used for all arithmetic and bit manipulation operations, as well as being used for most data transfer functions.

The X Index Register (X)

The X register is one of a pair of "index" registers. It is like the accumulator, in that it is an 8-bit register, but only the most simple operations can be carried out on X. Its primary use is as an index for indexed addressing modes — See the section on addressing modes.

The Y Index Register (Y)

The Y register is identical to the X register, also being used mainly as an index.

The Stack Pointer (SP)

The stack pointer is another 8-bit register, which contains a pointer to the current byte on the 7501 machine stack. See the section which follows on the stack for an outline of its use.

The 7501 Processor Stack

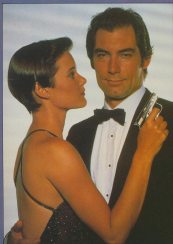
In Basic, you have two instructions to alter the flow of a program — GOTO and GOSUB is simply — it just performs an absolute, unconditional jump to a line. GOSUB does the same thing except that before the jump is performed, the statement to which the program should return is first pushed into an area of memory called the Basic stack. Then, when a RETURN statement is encountered, the program pulls the return statement back off the Basic stack and jumps to it. Machine language works in the same way using its own machine stack, except that the GOSUB command is replaced by JSR (Jump to Sub-Routine). When JSR is executed, the address of the next command is pushed onto the stack so that when a subsequent RTS (Return from Subroutine) is found, the address can be pulled back off the stack and transferred to the PC to continue program execution. The process of pushing and pulling is automatic and controlled by the Stack Pointer. The machine stack is in fact a 256-byte (1 page) area of memory located at \$0100 onwards.

In addition to being used for sub-routines, the stack can also be used as a temporary store for numbers, avoiding the need to use a specific area of memory. The PRA, PLA, PHP, PLP instructions push numbers onto the stack and pull them back off it again. When a number is pushed or pulled, the stack pointer is decremented or incremented accordingly. You can however control the stack pointer contents directly using the TOS and TSS instructions. The stack pointer is a single 8-bit index into the stack area at \$0100.

Note that when using the stack, you must of course march every JSR with an equivalent RTS, and push with equivalent pop. The stack can accommodate 256 bytes, so up to 128 levels of nested sub-routines are theoretically possible. It is certainly unlikely that you will ever need more than this.

We've had to cut Mark short here, but you'll find more in the second installment.

A Serious Case of Bondage



Crashy — Kevin Costner takes a look at the latest Bond film and its accompanying game from Domark. After 27 years the world's best known secret agent — James Bond returns once again and, as the film posters state, "This time he's out for revenge".

Licence to Kill is the 16th official Bond movie (not counting Casino Royale and Never Say Never Again) and the second for new boy Timothy Dalton (not the last as some television networks would have us believe). Dalton himself is very pleased with 'Licence', "it's much closer to the original character in the books" he enthused at the recent premier in London's Leicester Square Theatre. "It's tough, tense, exciting, everything a Bond film should be". It's certainly a lot tougher than previous films which has been the main reason for Licence to Kill's 15 certificate (the first time a Bond film has received anything other than a 'PG' or equivalent). This was only obtained after over a minute of the more violent moments were cut.

However, this doesn't detract from the film's enjoyment, which is the main thing really. It's just a shame that our film censors in this country aren't a little more like their counterparts on the other side of the Atlantic who have given it a PG13, this means that anyone can see it with a parent but you have to be 13 or over to see it on your own. Seems reasonable enough to me!

The film itself centres around the twilight world of Drago Baron Frank Sanchez — an unsavory character who escapes imprisonment and gets his own back on those that caught him (Bond and Felix Leiter). New Bond becomes a vigilante to get Sanchez back for what he's done, but he must do it without the backing of her Majesty's Secret Service. This time he's on his own.

All the familiar elements are there, evil haddies, beautiful girls, spectacular stunt sequences and of course those ingenious gadgets courtesy of 'Q', played for the 14th time (he wasn't in *On Her His and Her Day*) by Desmond Llewellyn. However all these elements have a much harder edge. One could almost be mistaken for thinking this was a Dirty Harry



film, particularly if one is used to the Roger Moore style of Bondry. If that's what you've been brought up on then prepare yourself for a shock... for this is a Bond for the 90's. The action is that bit more violent, the villains are that much more realistic (which makes them all the more sinister) and the Bond girls aren't just the stereotypical Bimbos of the past, they now have quite an effect over the action. Is it welcome? Most certainly. As Dohen puts it "This is an Action, Adventure Thriller for adults that kids can also enjoy". A well written plot from the pen of Michael G. Wilson and Richard Matheson, competently directed by John Glen, this Ekk Bond

Directorship, makes for an excellent entertainment.

It is nice to see Demarcq has pulled out all the stops to release the game to coincide with the general release of the film. This will be Demarcq's fourth game based on a Bond film. The first was View to a Kill back in '85, followed by The Living Daylights and Live and Let Die. It'll be launched across all formats, including the Commodore 64, Amiga and PC formats, simultaneously. All three versions of interest to Commodore owners look and play really well, particularly the Amiga version.

The game comprises six parts, split up among three screens from the film.



The first involves James and Felix flying a helicopter at low level over Gray Cay, avoiding tall buildings and gun emplacements... a little like Thunder Blade. Once through that, you (as Bond) get off on foot after Sanchez but you have to get through plenty of his henchmen first. Weaving your way through a dangerous maze of exploding oil barrels and bullets, you must get back to Felix in one piece. Then it's back in the air to capture Sanchez's plane. Here you are forced onto the back of the plane where you must attach a towrope and bring Sanchez in.

In Scene Two 007 must foil a narcotics drop, but first he must survive in the water armed with nothing but a knife against divers and armed boats. Bombs are added for destroying the Drugs caches as well as saving with your life. Next, you attack yourself, via Harpoon's a seaplane's pontoons for a high speed, harlequin water-ski chase, avoiding catamarans whilst you work your way to the seaplane and gain control.

In the final scenes you are in hot pursuit of Sanchez having just obli-



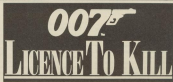
located his processing plant. He's taken to the winding mountain paths as part of a conveyer of 18 wheel juggernauts. The end is near, the focus of your vengeance is just around that bend. You just have to destroy each of the tankers one at a time until you get to the one at the front of the conveyer with you know who in it, but because he's packing some serious heat in the form of Stinger missiles.

The whole game takes the form of a seriously scrolling, birds-eye view shoot 'em up and works very well as a game. Which as a movie die-in is rare indeed.

Bond Filmography

Dr No - (1962) Sean Connery - Dir Terence Young.
From Russia With Love - (1963) Sean Connery - Dir Terence Young.
Goldfinger - (1964) Sean Connery - Dir Guy Hamilton.
Thunderball - (1965) Sean Connery - Dir Terence Young.
You Only Live Twice - (1967) Sean Connery - Dir Lewis Gilbert.
On Her Majesty's Secret Service - (1968) George Lazenby - Dir Peter Hunt.
Diamonds Are Forever - (1971) Sean Connery - Dir Guy Hamilton.
Live and Let Die - (1973) Roger Moore - Dir Guy Hamilton.

The Man With The Golden Gun - (1975) Roger Moore - Dir Guy Hamilton.
The Spy Who Loved Me - (1977) Roger Moore - Dir Lewis Gilbert.
Moonraker - (1979) Roger Moore - Dir Lewis Gilbert.
For Your Eyes Only - (1981) Roger Moore - Dir John Glen.
Octopussy - (1983) Roger Moore - Dir John Glen.
View To A Kill - (1985) Roger Moore - Dir John Glen.
The Living Daylights - (1987) Timothy Dalton - Dir John Glen.
Licence To Kill - (1989) Timothy Dalton - Dir John Glen.



128 Corner



Pin	Type	Note
1	VSYNC (VNC)	Luminance (VNC) output
2	GRND	
3	AUDIO OUT	
4	VIDEO OUT	
5	AUDIO IN	Composite signal output
6	COLOR OUT	
7	NC	Chroma signal output
8	NC	No connection



Pin	Signal
1	Ground
2	Ground
3	Red
4	Green
5	Blue
6	Intensity
7	Monochrome
8	Horizontal Sync
9	Vertical Sync

Dear 128 Corner:

Yes, I own a Commodore 128 and have done since I upgraded from the good old 64 back in 1986. I was pleased to see that you have decided to provide a forum in your magazine for this excellent machine.

I agree with your views regarding the poor availability of software. I suppose the reason for this is that the machine is too small to be a fully

fledged business machine and too large to be a game machine. However, by viewing the magazines it is possible to get hold of some magnificent software that is specific to the computer.

I was surprised that you missed out the C128 version of *GEOFFRATTE FORKSAWOP* as an example of a reasonable word-processor. I was initially put off by the GEOS system

because it did not allow printing in NLQ mode. The fonts that were provided were too 'dotty' as my Star 5610-C for serious letters. This has been solved in *GEOFFRATTE FORKSAWOP*.

The biggest advantage of the program is that you can take large sections of figures from *GUOCALC 128* and paste them into letters without having to write them out again.

For straightforward word-processing I would think that *SuperScript 128* is best. Its sister program *SuperBase 128* is another program worth seeking out if you want to create a database for any reason. The Basic like programming language, included in the program, allows you to manipulate the data in your own applications programs and adds a new dimension to using a database.

The hype surrounding the design makes everyone wonder if they shouldn't part with their old computer and buy a new one. I like playing C64 games on the 128 and am happy with spreadsheet, database and word-processor applications that I have for the machine. In the words of the Americans, if it ain't broke, don't fix it.

Colin Mervin, Bolton, Lancs

Have you tried Post Master 128 or SuperWrite II, both are very excellent applications. I actually agree with your comments about the 128. I too would like my own personal Amiga, but why have a Beowulf to learn to drive when a Mavis Africa will do the trick???

Dear 128 Corner,

It's good to see that you intend to continue covering the C128 computer. I've had one for a few years now and use it quite regularly for wordprocessing and filing. I must say that I prefer using the C128 over the 'real' computers that we have in the office, it's a

lot easier to use and more friendly (I can also play games on it when I'm bored).

When I first purchased my C128 I was very disappointed to find that I couldn't use the 80-column video mode without purchasing an expensive colour monitor, which I couldn't afford. I was intrigued however when I saw people advertising switches that allowed you to use 80 column mode on a cheap monochrome monitor - not as good as colour but certainly better than using 40-columns all of the time. I investigated this further and careful examination of the 80 column video socket in the manual revealed that there is an 80 column composite video output on pin 7 of the connector. Converting a suitable plug to this pin and the earth (pins 1 and 3) means that the computer can be plugged into a suitable composite monitor.

If you want to use the same monitor to display 40 column video then this can also be achieved. Pin 4 on the 40-column video connector is

the video out and pin 2 is ground. Connect the video and ground to a suitable connector as before and you can use the monitor to display 40 columns as well.

I know of a number of other people who use a composite monitor with a lead as described above, in some cases they use pin 1 (LUMI SYNC) instead of pin 4 though on my monitor the results aren't as good.

I have taken my lead even further and placed a double throw switch in in the lead. One pole of the switch goes to the 80-column video pin, the other pole the 40-column video pin. The central pin of the switch is wired to the video plug for my monitor and all of the earth pins are connected together. Switching between 80 and 80 columns is now a simple matter of pressing the 40/80 switch on my monitor and flicking my own switch.

One further advantage of using a video lead such as the one described above is that you can, contrary to popular belief, get 80-columns on your

television, as long as you have a video recorder anyway. Most video recorders have an input for a video camera. This input is a composite video one. Plug the lead into this socket, make sure that your TV and video are set up to display from the camera input and you can see your C128 in 80-column mode on your TV. The quality may not be as good as a monitor, but at least it's 80-columns! S. Garton, Rotherham, S. Yorks.

Get In Touch

C128 Corner is a forum for all 128 users. If you have any comments, suggestions or questions do send them in. Without your contribution then C128 Corner will not be able to continue, so come on, write to:

C128 Corner

Your Commodore

Argus House

Boundary Way

Hemel Hempstead

Herts

HP1 1ST

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema.

Most of them comply with the rules contained in the *British Code of Advertising Practice*.

But some of them break the rules and warrant your complaints.

If you're not sure about which ones they are, however, drop us a line and we'll send you an abridged copy of the *Advertising Code*.

Then, if an advertisement bothers you, you'll be justified in bothering us.

The Advertising Standards Authority. If an advertisement is wrong, we're here to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

The space is donated in the interests of high standards of advertising.

FORM MASTER

Don't just believe us about Form Master - here are some of our testimonials.

"I have purchased hundreds of systems over the last forty years and Form Master is by far the most brilliant system. James Chapman is an outstandingly polite and calm man. My profit from date of purchase is £1,500".

B. Manchester, *Leeds* 10.5.88

"Form Master (they) when I contacted it it and finds good priced adverts in comparison to other markets."

P. Jones, *Cardiff* 1.1.88

"This first time I used Form Master I had a 12.7 minute and I have never been so happy. I am currently showing a profit of £1,000. James (he) is averaging 80% winners and makes their winning devices in 10 seconds. I won't be without it."

P. Smith, *Sheffield* 15.1.88

"I had a number of winners within a week of using the system and in the last week alone I have not placed a losing bet. You can imagine how delighted I am."

B. Newton, *Walsley* 10.5.88

Form Master has averaged a 10% success rate over the National Hunt Season to date. Continued figures. Don't be in the losing end for the last season - contact Form Master now.

1 SPECIAL OFFER for singles (14 days) 1st Prize and 30 Annuitant packages. Only £25.00 per VLT & P&P

* Available for Commodore 64, 128, 128+, 128C, 128C+, 128C+, 128C+, 128C+

ORDER FORM MASTER about from James Chapman using any major credit card or by cheque.

To order by Credit Card simply call 0434 510000 and an order form will be sent to you by post.

For more information call 0434 510000. Post and Packing.

Reply Now

NAME BY LAST

First Name

Second Name

Third Name

Fourth Name

Fifth Name

Sixth Name

Seventh Name

Eighth Name

Ninth Name

Tenth Name

Eleventh Name

Twelfth Name

Thirteenth Name

Fourteenth Name

Fifteenth Name

Sixteenth Name

Seventeenth Name

Eighteenth Name

Nineteenth Name

Twentieth Name

Waiting for

Dave Hughes and James Sewell explore the world of the masked avenger, who is considered by some as one of the five most recognised figures in the world.

The Batman, the Dark Knight, the Caped Crusader – call him what you will, his fictional presence has played a major role in our subculture since his conception in a 1939 issue of *Detective Comics*. Fifty years on the masked vigilante, cloaked and hidden in the dark shadows of a fake New York metropolis known as Gotham City, is about to embark on his second big screen adventure. And this time it's for real.

It is ten years since Warner Brothers, movie-making partner company of Batman's publisher DC Comics, first considered the idea of a new Batman feature film. Since that time, comics like *Watchmen* and *The Dark Knight Returns* have brought a new era to the graphic story genre, adding political, racial, sexual and social issues to a field previously associated with muscle-bound, costume-wild freaks of nature. And *Batman: The Movie* is suddenly big business. Unlike the Caped Crusader's previous screen image, Sam Hamm's original 1986 screenplay concentrated on keeping Batman true to the comics – like Howie Kanin's original character, he is a dark brooding vigilante forever seeking vengeance for his dead parents, rather than the over-the-top Adam West/Bat Ward team of the camp mid-Sixties spoof series. This time, POW! and ZAP! are left redundant by a movie that pulls no punches.

For this movie, Batman trades in his blue-and-grey hooded for a matt black rubber outfit resistant more of Frank Miller's Batman than Adam West's. This time, no-one will be left smiling at the pranks of his arch-enemy the Joker. And we all know that Batman's boyish sidekick Robin isn't even in the picture – due mainly to the expression of the part of the Joker which followed Jack Nicholson's reluctant acceptance of the role ("It's like God created him for the role" producer Mike Uslan is quoted as saying). Replacing director Tim

or a Dark Knight

Batman directs, and Beetlejuice himself, actor Michael Keaton, fills the Batman's subterranean shoes. Aside from Jack Nicholson, Jerry Ball and Kim Basinger both appear, but Robin the Boy Wonder is not set to appear until *BATMAN 2* (already being scripted), about which movie remains already flying. Including the possible casting of Robin Williams as rhyming rival the Riddler... Originally, however, Sylvester Stallone and Arnold Schwarzenegger were considered for the title role, both being rejected because they did not fit the image of Batman's millionaire alter-ego Bruce Wayne. Also, as Tim Burton says, "the idea of Arnold Schwarzenegger in a Batman costume is frankly ridiculous!" Instead, Michael Keaton wears built-up shoes, a latex skin and false hair inside his padded suit, further enhancing his transformation from playboy Wayne to Dark Knight.

The plot, like the rest lot, has also undergone major surgery. Sam Hammet's first draft featured The Flying Graysons and their young offspring (Robin) who, like the young Bruce Wayne, witnesses his parents' murder, this time at the white-gloved hands of the Joker. The friendly Scrooge image of the Batmobile has been replaced by a Gothic monstrosity which wields a machine gun and is powered by a jet turbine—a fast and powerful reminder of the Batman's might. Gotham City, depicted in the new film as its biennial celebrations, is a grimier, darker vision of New York. Commissioner Gordon is fighting police corruption as well as the Joker-dominated underworld collaboration, and the street is filled with the dark and sinister images of wrecked cars, sevens and drug pushers, invading the well-to-do areas and making for a violent and dark base for Batman's comeback.

The Joker is depicted as a right-brain madman, driven insane by his own disfigurement following a chemical accident, and seeking his vengeance on humanity. Batman is the only thing that stands between him and his evil destiny, and the film follows the interaction between two society muffs, each infatuated with his role and each closer to the other than either

would care to admit—a plotline linked closely with British writer Alan Moore's recent novellas *The Killing Joke*. Closer still to the movie's plot is Frank Miller's highly-acclaimed *The Dark Knight Returns*, but a crucial scene was (in true Hollywood style) cut from the movie following news that Warner had agreed to a sequel in the cinematic confrontation of Frank Miller's story. Batman only just stays short of throttling the Joker to death, at which the Joker, in his last and best punch line, breaks his own neck so that Batman will take the rap for his death; this scene was written out following Jack Nicholson's agreement to appear in a sequel... The high Gotham City set—the largest film set since Cleopatra—stands empty at Pinewood Studios in Buckinghamshire, waiting for the crew to return in February for more adventures of the caped crusader. The movie's success is already secured.

At the time of writing *BATMAN* is two weeks away from its U.S.

release, but already scores of people have gotten hands-on. Fans have been reported as seeing *THE DEAD END*, just to catch the smug-second trailer for *BATMAN* shown along with it. Despite Jack Nicholson's opinion that the movie is "stunned", the media coverage and the secrecy in which the movie has been shrouded have made a sure-fire winner for Warner. The *Batman* comic series The Cuh has rocketed to 2,500% of its cover price virtually overnight. Illegal sniffs, scripts and unlawfully shoddy pirate video copies are changing hands at massive prices. The merchandising from the series is likely to cover the cost of the 136 million movie in a matter of months, with Warner running the biggest merchandising campaign since *STAR WARS*, with novels, action figures, masks, model kits and even credit cards aping the Batman of the Sixties.

So what is it about this masked hero that demands such slavish devotion?





tion from his legions of fans? Perhaps it is that, unlike irradiated or miracle-endowed superheroes from other comics, Batman is one of us - an ordinary man with no super-powers, who spends two years of his life building himself up to fight crime and avenge the violent deaths of his

parents, ensuring to stamp out injustice to the best of his ability. The simplicity of the character's origins make him easy to identify with, and more believable in a fantasy universe where an orphan from space wears his underpants outside his tights, and where the mere bite of a spider

transforms a man into a superhuman... He is fallible, a haunted, emotional lover constantly at odds with his hidden identity, and the film follows his inner turmoil and endless bereavement for his lost childhood.

After all, at the end of the day, it's about time Batman came of age...





The Music

"I've seen the future and it will be..." is the first lyric of the first track on the sound track album. And what a fitting (and apt) lyric to kick off the Batman handwagons.

Although there are nine individual songs on the album, only four were to be written, and this record only encompasses half of the music in the film.

Danny Elfman (credited for his score in *Beetlejuice* and *For West-Beag-4*) provided the "tuned" music throughout. *Batman* doesn't seem to mind though, his presence was requested by Jack Nicholson (the Joker), and being signed to the Warner record label meant he was the obvious choice anyway. It seems as though Elfman has made this his album for 1989, putting more effort into each song than you could expect from a normal soundtrack composer.

The songs are well constructed, the album is cunningly arranged, with a mix of several different tastes, and the flavor of each track is unmistakably Prince... With one exception. *Batman* is probably the weakest track, not being altogether funky nor moving with Sex - as

many Prince songs before it - just setting for plain old dance music. But, as Prince states: I have seen the future and it will be... more *Pentecost Park* soundtracks.

The Film

Skeptical Batfans may have dismissed the movie before its release, and who wouldn't with the amount of false rumours spread about in the tabloids. But now, upon its release in America, there seems to be little alternative to congratulating the entire team involved.

Michael Keaton is superb as both Bruce Wayne and the Dark Knight Detective, and wanda fail to explain Jack Nicholson's performance as the cranky, but always dangerous, Joker. Page Klee Scragger (Vicki Vale) offers one of her best roles.

On its American premiere, *10,000* dedicated Batmanians appeared to see the stars parade before them. The lucky few who were able to purchase tickets could not shake their eyes from *Sylvester Stallone* (among other stars), and cry and cheer their way through a truly stunning movie.

Once the film has broken all box office

records, as it promises to do, and hit music goes, young and old, staggering out of the cinema exhausted, we will have to contend with the fact that it will not be too long before *Batman 2, 3, 4* and many more will come our way to keep our addiction satisfied. Long live Bruce Wayne, long live the Joker, long live, long live, long live Batman...

The Game

There are constantly in the light of holding their tale, restricting their output, pooling their resources, and pulling a major license outside out of the bag. There can be no better licenses to deliver than *Batman* - the movie, and we are to be sure that it is a winner of the tag to the bank.

The game is split into four sections, all based upon scenes in the film. The first is based in a Gotham City chemical factory (where the Joker is created), which is viewed from a side perspective. One feature in this installation at least are in *Batman*'s rope which he can use to swing between levels - something akin to *Acrobatic Commandos*.

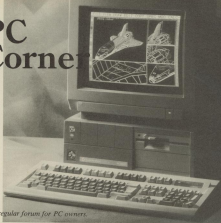
The second subgame revolves around the Batmobile, which Batman must drive in order to find the Joker's van. Not only must you avoid other traffic, but you can use your batwing to turn corners at a high velocity.

Section three also features one of Bruce Wayne's modes of transport, this time the Batwing, mounted for. The Joker plans to wipe out Gotham using balloons full of deadly gas. As Batman, you must fly along and cut the wires keeping the balloons afloat to be ground, thus eliminating the danger.

The fourth section is very similar to the first, except the Dark Knight detective has to visit a church, for the final confrontation with the Joker.

Given predict that the game will be released at the same time as the film in this country - August 1989 - so it should prove to be another step in the merchandising boom this autumn, and almost certainly the biggest license in the software industry, ever.

PC Corner



Our regular forum for PC owners.

I have come to rely on my computer a great deal in the course of my day to day life. I have three machines, all of which are made by different manufacturers, all are "clones" of the IBM PC design and none of them were built by IBM. I use three computers to write software, to do graphics and text for manuals and adverts, control my recording studio, keep up to date with various things and people across three continents and keep my financial records in order for the pleasure of the VAT and tax man. I even sometimes play games on one of them. Before the dawn of white coats come to take me away I'd like to explain why.

The great strength of the IBM PC family of computers is their flexibility, this is in part due to the availability of a great number of different computers and expansion cards and the sheer volume of software available. The immense number of manufacturers in the PC market means that there is a large spread in the price and power of the hardware. The wide range of expansion cards available means that you can use a PC to perform almost

any task by simply adding a relatively simple (and therefore cheap) printed circuit board to the computer. As with the hardware, having lots of software for the machine gives you the advantages that competition within any field will give you - quality and diversity of products.

Another important aspect of the strength of the IBM PC market is that it will have a long life, even IBM with the release of their PS/2 range of computers haven't been able to put a significant damper on the availability and innovation within the PC field. It's also to know that when you buy a PC that next month or next year you won't be left out on a limb with a computer that has no support or new software.

The speed at which prices are coming down is also amazing, for instance you can now buy a powerful 80386 based AT style PC from New Computing with 1 megabyte of RAM and a 40-megabyte disk for around £1400 (plus VAT) which is what I paid for my 28-MBps XT/PC two years ago. There is also a pocket sized PC being released by Atari soon, which

uses memory cards instead of disks and has a serial port for RS232 communications and a lap-top PC from Farnite designed for professional music applications.

Words for the Whys

The IBM PC is basically a "vanilla" computer, it doesn't have the design compromises that are a feature of a lot of home computers. Many competitors have built their computers designed with a particular goal in mind, this means that they tend to be low on graphics, or extremely low cost but have some special features.

The upshot of this is that the basic PC doesn't perform any particular task exceedingly well, but can perform most tasks competently. The success of the PC design is really quite obvious. Just look at the large number of relatively cheap expansion cards on the market, while a lot these are aimed at the business or industrial market there are a few which are aimed specifically at the "home" business as "home" use or can be used for amusement.

Possibly one reason for this success is the mechanical integrity of the PC system. While there is nothing particularly novel about the idea of having additional cards for a personal/home computer, the allowing of space inside the case for them means that the expanded system is robust enough to stand hostile environments (for example those involving pets and small children). The inclusion of the expansion bus in the PC's design also gives us an insight into the microcosm of the PC which in turn explains just why there is so much commercial and public domain software available for the PC.

The History Bits

The disk operating system (or DOS) used by the PC is based on one of the earliest systems designed for 8 bit microcomputer systems. This DOS was called CP/M, which stands for Control Program for Microcomputers, the major feature of this software was that it wasn't tied to any one computer. This meant that manufacturers - and even hobbyists - could build a computer for which there was a large amount of common software available. A number of popular PC packages were originally written for CP/M such as Wordstar, dBase II and Super Calc.

The main problem with CP/M was that it was plagued by the fact that there was no standard disk format, the main UK CP/M user group provides disks from its software library in over 120 different disk formats!

The Public Domain

The hobbyist involvement meant that there were a lot of dedicated technical people writing programs for their own use and/or amusement; these people got together in computer clubs in the USA and other parts of the world and released their software in the 'Public Domain' (or PD). These programmes were quite happy to let other people use their software and would often provide limited support, as long as the software wasn't sold commercially. Incidentally, the CP/M operating system isn't dead yet, the Amstrad PC/N series uses a version of CP/M version 2.2.

The original IBM PC's operating system was very closely related to an early version of CP/M, which had two flaws, the first was that people who wanted to upgrade their computer to a 16 bit processor could move to a

fairly familiar operating and software environment. The second result was that existing commercial packages could be easily converted (or ported) to the new machine, and since Microsoft, the suppliers of the PC's operating system, also released the DOS software as a separate product, other manufacturers could also build it in to their computers.

The User Groups that distributed the CP/M public domain software now also have MS-DOS programs in their catalogues which range from word-processors, games and CAD systems to the entire King James version of the Bible (8 disks). The sort of software available from the user groups can be very variable in quality, some of it is very good, some obviously not finished. I use a public domain communications package called Procomm, since it is superior to most commercial terminal emulation programs I've seen.

Operating System Enhancements

Mind you, you don't have to put up with the MS-DOS operating system, if you don't want to, there are various ways of 'improving' it. Like the PC hardware, the operating system can be enhanced by adding on what is sometimes called a 'front-end'. Examples of front ends used to improve the user interface of MS-DOS are the graphic environment managers such as Digital Research's GEM or Microsoft Windows and the menu driven systems such as *Novell* and *LDN*.

These programs alter the look and feel of the operating system to make it easier to use, either by implementing the WIMP standard (Windows, Icons, Mouse Pointing device) or by displaying the disk files in your current directory on the screen with a menu of standard commands. *Microsoft Windows* also falls into the category of programs that give your PC the ability to run more than one program at a time - the so-called multi-tasking. Other packages that give you the capability of *DOS* enhanced *Double DOS*.

The End PC

This column has just scratched the surface of the IBM PC compatible computer world. The aim is to give an idea of what's out there, thereby explain why this stodgy little computer has become so popular. In future columns I hope to go into greater depth on how

to get the PC to do various things, some of which it was not designed to do. I shall also look at some specific programs and expansion cards that I have come across and actually used, not so much as a review, but as a user's comment and as solutions to specific problems which I might have some access.

Organisations & Products Mentioned

This section is for reference, the prices are (where shown) exclusive of VAT and aren't necessarily the cheapest, they're just there as a guide.

Non AT386-28, PC/AT with 38686, 40 Mbyte Hard Disk, 1 Mbyte RAM
Price - £1395.
Supplier - Non Computing, 01-338 5810

YAMAHA CI - LCD Laptop PC/AT, 36 Mbyte Hard Disk, 2 Mbyte RAM
Price - £295.
Supplier - YAMAHA Sales, Condon Street, London

CP/M & MSDOS Users Group, 72 Mill Road, Hawley, Dartford, DA2 7E2

CompuLink User Group, Suite 2, The Sanctuary, Sarnham, KT6 4DU, 01-894 84

Procomm v 2.4.2, communications software
Supplier - CompuLink User Group

Microsoft Windows, Graphics environment for PC
Price - £59.
Supplier - PW Computer Supplies, 01-564 4048

View, DOS file and directory manager from Executive Systems Inc.
Price - £44.
Supplier - PW Computer Supplies.

EDM, DOS and directory manager from Edmark Inc.
Supplier - Quake, 01-894 2425

DESQview, multi-tasking environment
Price - £49.
Supplier - CompaAdd, 0400 335315

Double DOS, Soft Logic's multi-tasking system
Price - £24.95
Supplier - Corporate Software, 01575 5461

Have

YOUR

COMMODORE



*delivered to your door FREE !**

That's right, if you take out a year's subscription to Your Commodore we will make sure that it is delivered to your door each month at no extra charge*. Just fill in the coupon below and send it to the address given with a cheque, money order or credit card instructions to cover the cost of the subscription. We'll do the rest.

*Overseas subscription rates include postage.

UK	£15.00
EUROPE	£20.25
MIDDLE EAST	£20.50
FAR EAST	£26.40
REST OF THE WORLD	£28.00
in USA	\$42.00

Actual Rates on Request.

Send this form with your remittance to:
INFONET LTD,
5 River Park Estate,
Biller Lane,
BERKHAMSTED,
Herts. HP4 3HL,
United Kingdom.

Please continue my subscription to Your Commodore with the issue.

I enclose my cheque/money order for £..... / made payable to
ARJIS SPECIALIST PUBLICATIONS

Or Debit my Access/Visa number

Valid from to Signature

Name Address

Postcode

☐ New subscriber ☐ Subscription renewed/extension (delete as applicable)

Oops!



Make your programming easier with this package of extra commands for the dedicated Plus/4 user

By Mark Everingham

There are two problems with computers. The first is *temporal* distortion, and the second is *human fallibility*. If you are thinking to yourself "What the hell does that mean?" then don't worry - it's really very simple.

Temporal Distortion is the phenomenon by which time seems to become compressed in the presence of a computer. What this means in practice is that when you pop up to your Plus/4 for "five minutes before tea", you find that tea has long gone and that it's about time you were thinking about some breakfast. The answer to this is, of course, a competent alarm-clock, so OOPS! includes such a clock with advanced user-definable alarms and reminders so that you don't forget lunch or that vital doctor's appointment this afternoon.

The other problem with computers, which I mentioned before, is that although computers, cassette-

recorders and disk-drives are relatively reliable (Commodore's are notorious for being very reliable but equally slow), human beings are sadly not so dependable. It is all too easy to crash your Plus/4 or NEW the one and only copy of your latest million-seller arcade masterpiece. This event will send most rational human beings into a cry of OOPS! (Or similar four letter words).

In order to protect you from these awkward circumstances, OOPS! has an *onNEW-type* command and a unique automatic program-backup facility making it as difficult as possible to erase or lose an important program.

The features that OOPS! offers are now becoming more and more common as parts of programming languages on the latest generation of 16-bit micros such as the Commodore amiga, but have not until now been available for the BASIC users of 8-bit home-computers like the Commodore Plus/4.

The OOPS! Basic Commands

OOPS! in practice consists of a program which adds thirty new commands to the normal Commodore Basic Operating-System. This may seem a lot of a daunting prospect at first, but they are all easy to use, and give an idea of how comprehensive the OOPS! system is. The commands can be split into six sections, namely *Clock Commands*, *Alarm Commands*, *Data Commands*, *Saving Commands*, *Information Commands* and *Miscellaneous Other Commands*. In a moment I'll demonstrate the use of each of these, but first a short explanation of command syntax.

Each of the OOPS! commands is used in exactly the same way as any standard Commodore Basic commands. Each can be used in Basic Direct mode or in a Basic program, and all can be abbreviated as explained later. Because the commands behave like normal Basic commands, any syntax they have can be replaced with a string variable, for example both the below have exactly the same meaning (to set the 24-hour clock to 12:38:29).

```
SETCLOCK "12:38:29"
```

```
CLS="12:38:29":SETCLOCK CLS
```

This property of the commands is useful when writing your own programs, for instance if you can't remember how to work the **SETCLOCK** command, you could write a program which would ask you for the Hours, Minutes and Seconds, and then set the clock for you using something like **SETCLOCK HH-MM-SS** ("MM" = *55 with the hours, minutes and seconds stored in HH, MM and SS respectively). This facility of the command means that you are not just limited to direct literal commands.

The OOPS! Clock Commands

The OOPS! system includes a 24-hour clock which does not interfere with the normal Basic 118 clock and can be made to appear on-screen all the time, whatever else you are doing with your Plus/4 so that you can run programs, print documents, display the disk-directory or whatever you want - the clock will keep running. Below are the commands used to operate the clock.

```
SETCLOCK "HH-MM-SS"
```


The **SETCLOCK** command is used to set the initial time of the 24-hour clock, as you will obviously not always start using the OOPSI clock at the same time of the day. Its syntax is the time enclosed within inverted commas. **HH** is the time under the 24-hour system, **MM** is the number of minutes and **SS** the number of seconds. Each should be separated using a single colon. The command can yield the error message shown below which can all be trapped using the normal Basic **TRAP** command, and are displayed as normal.

SYNTAX ERROR - You have omitted the time from the command.

ATYPE MISMATCH ERROR - You have typed a numeric parameter instead of the time string.

NUMERICAL QUANTITY ERROR - You have entered a time string which is either not of the form "HH:MM:SS" or has a number in it outside the valid range 0-23 for hours or 0-59 for minutes and seconds.

CLOCK ON/OFF

Once you have set the time on the 24-hour clock using the **SETCLOCK** command, you will want to be able to see what time the clock is showing all the time like a normal alarm clock. To do this just type **CLOCK ON**. From then on the clock is always displayed in the top-right corner of the screen. Note that the OOPSI system cannot be used with the high-resolution graphics screen, so the clock is always visible whenever you are doing. However, there may be times when you don't want the clock display to be visible, for example when writing a program that makes use of the full screen. To remove it from the screen type **CLOCK OFF**. Note that although the clock will now be invisible, it is still running and will still display the correct time when a subsequent **CLOCK ON** command is executed. The **CLOCK** command gives only one error message - **SYNTAX ERROR** indicating that you have left out, or possibly replaced the **ON/OFF** argument.

START and CSTOP

The **START** and **CSTOP** commands are used to respectively start and stop the OOPSI clock. Their main use is for accurate setting of the clock. For example if the time were coming up

to 10 o'clock you could use the program below to set the clock very accurately:

```
10 CSTOP
20 SETCLOCK "10:00:00"
30 PRINT "PRESS A KEY WHEN
  TIME IS 10 O'CLOCK"
40 GETKEY K$
50 CSTART
```

Although it is not necessary to stop the clock while setting it, these commands are useful for setting the clock very accurately or taking the clock as a timer started by pressing a key like the button on a stopwatch.

The **CSTART** and **CSTOP** commands require no arguments and so produce no error messages.

The OOPSI Alarm Commands

If you have anything important that you don't want to forget to do during the day, you can use the OOPSI alarm section of the clock to remind you to do this. As well as giving an audible warning, the alarm can also be made to display a message to remind you exactly what the alarm is set for.

SETALARM "HH:MM"

The **SETALARM** command is used to set up the time at which the alarm will sound. It works in the same way as the **SETCLOCK** command but the alarm time is expressed in only hours and minutes and not also seconds. The error messages the command gives are identical to those for the **SETCLOCK** command.

WARNING "Alarm Warning Message"

I have mentioned before that as well as making a noise when the alarm time is reached, OOPSI can be made to display a message as a reminder. This message appears inverted and flashing in the top-left corner of the screen to draw attention to itself. The **WARNING** command is used to set up this message (or when the alarm goes off, the message should be enclosed in inverted commas and be not more than 25 characters in length. The message may be made up of all alphanumeric characters except control codes such as **BS** and **BSOFF**. The error messages that the **WARNING** command yields are shown below.

SYNTAX ERROR - You have left out the warning message.

ATYPE MISMATCH ERROR - You have entered a numeric argument in place of the warning message.

STRING TOO LONG ERROR - You have entered a warning message that is more than 25 characters in length.

ALARM ON/OFF

Now that you have set up the alarm time and message, you can switch the alarm on using **ALARM ON**. If you wish to turn the alarm off to reset the time and message, use the **ALARM OFF** command. When the alarm time is reached, a warning sound is made and the warning message displayed on the screen. There are three ways of turning the alarm off. The first way is by typing **ALARM OFF**. The other two methods are described in the **OK** command section listed under Events. The **ALARM** command gives one error message:

SYNTAX ERROR - You have omitted the **ON/OFF** argument.

The OOPSI Event Commands

What are events? Well they are just what they sound like - events, occurrences or happenings during the day. Still none the wiser? Well, most of us find that we have more than one thing that we want to remember to do on one day, or a single alarm is of limited use. For this reason, OOPSI has eight 'Event Alarms'. Each of these events works in the same way as the main OOPSI alarm and may be programmed with an event time and associated message. Within a single day you can get your Plus/4 to remind you of up to eight events, enough for those of you with the most densely packed social diaries!

SETEVENT EVENT #, "HH:MM"

The **SETEVENT** command works in the same way as the **SETALARM** command, but is used to set the time at which an event alarm will sound instead of the main OOPSI alarm. Its syntax consists of the number of the event you wish to set, in the range 1-8, followed by the event time enclosed in inverted commas and separated using a comma. The time is entered in an identical format to the **SETA-**

ALARM command. The command gives similar error messages, which are shown below:

Syntax ERROR - You have omitted the event number, event time or separating comma.

TYPE MISMATCH ERROR - You have entered a string instead of the event number or a number instead of the event time.

ILLEGAL QUANTITY ERROR - Either the event number you have specified is beyond the range 1-8, or you have entered an illegal event time.

MESSAGE EVENTS, "Event Message"

As mentioned before, each event alarm can be setup with a corresponding warning message in the same way as the GPPS main alarm. The **MESSAGE** command is used to setup one of these messages. It takes as its arguments the event number in the range 1-8 and the message that you want to be displayed when the event time is reached. The message may be up to 23 characters long and should be enclosed in double-quotes and preceded with a comma as shown above. An event message is displayed flashing and inverted in the top-left of the screen in the same manner as the alarm warning, and using the **WARNING** command. The **MESSAGE** command gives the error messages that follow:

Syntax ERROR - You have omitted the event number, event message or separating comma.

TYPE MISMATCH ERROR - You have replaced the event number with a string or the event message with a number.

ILLEGAL QUANTITY ERROR - The event number you have entered is outside the range 1-8.

STRING TOO LONG ERROR - The event message you have entered is longer than the allowed 23 characters.

ENABLE EV = 1, EV = 2, ..., EV = 8
Once you have setup an event with both event time and event message, you need some way of switching-on or enabling the event, so that when the event time is reached the correct sound is made and message displayed.

This is done by means of the **ENABLE** command.

The **ENABLE** command has a flexible syntax in that it can take one or more parameters rather like the Commodore Basic graphics commands. The basic argument is the number of the event that you wish to switch on, or enable. Thus to enable event number 4, just type **ENABLE 4**. If you want to enable more than one event, for instance events 1-4, it is a bit of a pain to have to type **ENABLE 1; ENABLE 2; ENABLE 3** etc. so instead you can simply type **ENABLE 1,2,3,4** saving down the amount of typing needed vastly. Each of the event numbers should be separated from each other using a comma, and be in the normal range 1-8. Note that the event numbers can be specified in any order, so **ENABLE 1,2,3** has the same effect as **ENABLE 3,2,1**.

When an event has been enabled, at the specified time, the event message will be displayed in the top-left of the screen as explained before, and an alarm will sound. The event alarm is much less harsh, and gentler than the main alarm sound as they are far reminders that are not going to be such a matter of life and death as the main alarm. So if you are of a nervous disposition - use the event alarms instead of the main alarm as they are less of a strain on the brain! The event message that the **ENABLE** command yields are shown below.

Syntax ERROR - You have either left out all the event numbers, or finished the line with an unnecessary comma.

TYPE MISMATCH ERROR - You have replaced at least one of the event numbers with a string instead of a number.

ILLEGAL QUANTITY ERROR - At least one of the event numbers you have entered is beyond the allowed range 1-8.

DISABLE EV = 1, EV = 2, ..., EV = 8
The **DISABLE** command performs exactly the same function as the **ENABLE** command, but has the effect of switching-off or disabling an event. Its syntax is identical to that of the **ENABLE** command and the error messages it gives have the same meanings as for the **ENABLE** command.

Alarm Priorities and "Shutting Them Up"

Some of you will have realised while reading the explanations of the GPPS! alarm and event commands that it is perfectly possible to set up all eight events and the main alarm to go off at the same time. What would happen if you did that? After all you can't make 9 inches and display 9 messages all at once, can you? The answer is NO, of course not, so a solution is reached in the form of alarm priorities.

Each event and the main alarm has a priority over the others. The alarm comes first, followed by the eight events in numeric order 1-8. To understand this, imagine that you had setup the alarm and events 1,3,5,7 all to sound at 10 o'clock. Because the main alarm has the highest priority, the alarm noise will be made and the alarm message displayed instead of all the others. "OK," you say, but if I can only hear and see the alarm, then what about events 1,3,5 and 7 - they might be important, too! To find out, read on...

OK (CTRL-6)

When an alarm or event goes off, there are several ways you can turn it off:

- If it is the alarm, you can type **ALARM OFF**
- If you know which event is occurring you can type **DISABLE EVENT #**
- The **OK** command

Methods (a) and (b) are all right, but a bit messy and, if you don't know which event is sounding, then you will have to try each one of the eight in turn to find out. The solution to this is the **OK** command. This command simply switches off whatever is occurring at the moment - the main alarm or the current event. Using this command, you don't have to know which event is occurring, and it also saves you quite a bit of typing.

I discussed earlier the question of what happens when more than one alarm or event is occurring, meaning that you can only see the one with the highest priority - "What about the others? Well, when you enter **OK**, it switches-off the event which is currently displayed on screen, and then displays the next one down the priority scale. For instance imagine that the alarm and all the events were set for the same time. When that time was

crashed, you would see the alarm message. When you type **OK** return the alarm is switched off and then event 1 is displayed. If you type **OK** return again then event 2 is displayed and so on until all the events have been acknowledged, whereupon the message in the top-left of the screen is erased and the sound turned off again. If you only wanted one alarm, you could use each of the eight events to give you a message of up to 328 characters, using the **OK** command to read the next section of message, with all events set for the same time.

For those of you who find typing two characters and pressing **RETURN** a bit tedious, you can achieve the same effect as typing **OK** by pushing **CTRL-O**.

For those of you who find typing two characters and pressing **RETURN** a bit tedious, you can achieve the same effect as typing **OK** by pushing **CTRL-O**.

OOPS! Saving Commands

The third important section of OOPS! is its automatic saving facility. Using this function, you can get your Plus/4 to automatically save a backup copy of the current program onto disk, so that you don't have to intrude your nose too far if you make a disastrous mistake. You can also make OOPS! mark each file it saves so that you have running log of different versions of your program on disk.

SETSAVING MINUTE

As mentioned before, OOPS! can be made to automatically save backup copies of the program that you are working on, so you need to tell OOPS! when it is to save these backup copies. For instance, if you are writing an important program, you may want it saved every five minutes (just to make sure that you don't lose it, but, if you are writing a long program, the saving takes a long time so you may want it to be saved every fifteen minutes or half an hour instead. The **SETSAVING** command is used to specify how often OOPS! should save backup copies. Its only argument is the period of time between successive savings, expressed in minutes. So, if you want your program saved every ten minutes, **SETSAVING 10** will set this up. You can use any time period between every one minute and every hour (60 minutes). The **SETSAVING** error messages are as follows:

Syntax ERROR - You have omitted the **MINUTE** argument.

TYPE MISMATCH ERROR - You

have replaced the **MINUTE** argument with a string.

ILLEGAL QUANTITY ERROR - The time period you have specified in minutes is beyond the range 1-60.

DEVICE DEV#

For those of you lucky enough to have more than one disk-drive attached to your Plus/4, OOPS! includes the ability to cope with this. When you are writing a program, you might want to have a utility disk containing an Assembler or Font Editor etc. in disk-drive 8, and an OOPS! backup disk in disk-drive 9. To do this, you would simply type **DEVICE 9** - the **DEV#** argument being the device number of the disk-drive on which OOPS! could save backup files. The device-number can be in the range 8-11 for four disk-drives, but you cannot use device 1 (The Dataset) as this is really too slow for automatic saving. The **DEVICE** error messages are shown below.

Syntax ERROR - You have omitted the device number argument.

TYPE MISMATCH ERROR - You have replaced the device number with a string instead of a number.

ILLEGAL DEVICE NUMBER ERROR - The device number you have specified is outside the valid range 8-11.

FILENAME "Filename"

When a backup copy is saved onto disk, it will obviously need to be given a name on the disk, and the **FILENAME** command is used to specify this name. The name argument should be enclosed in inverted commas and may be of up to ten characters in length. It may be made up of any alphanumeric characters but may not include spaces. If you do include a space in the name, the actual name will be truncated by the space character. For instance, entering **FILENAME "HALLO THERE!"** results in a file-name of "HALLO". The **FILENAME** error messages are as follows:

Syntax ERROR - You have failed to enter a file-name.

TYPE MISMATCH ERROR - You have either replaced the file-name argument with a numeric argument or have not enclosed the name in inverted commas.

STRING TOO LONG ERROR - The file-name you have entered is longer than ten characters.

OOPS! Backup File Marking

It is a little pointless to keep on saving backup copies of a program onto disk unless you know what each file contains. For example, if you were saving backup copies every two minutes and each of them were called simply "PROGRAM", you might have some files which were saved two hours ago when you were just starting the program, and some which had only just been saved. Obviously you need to know which is which! OOPS! offers several ways of marking your programs, or you can just choose for the last backup copy to be saved every time a new copy is saved. This section discusses the various commands for using the file-marking facility.

VERSION VER#

The first way of marking a backup copy saved by OOPS! is with a version number indicating at what stage of development that backup-copy is. You can tell OOPS! which version number of the program you want the first backup copy to be marked with by using the **VERSION** command with **VER#** being the version number. Thus if you want to start with version number 43, enter **VERSION 43**. OOPS! can cope with version numbers in the range 0-999, and this should be quite sufficient unless you are in the habit of saving 1000 backup copies of programs! The **VERSION** command gives the following error messages:

Syntax ERROR - You have omitted one of the **VER#** arguments.

TYPE MISMATCH ERROR - You have replaced the version number argument with a string.

ILLEGAL QUANTITY ERROR - The version number you have specified is outside the valid range 0-999.

NUMBERMARK

This is the first of the commands to tell OOPS! how to mark backup copies of a program. The **NUMBERMARK** command makes OOPS! add the current version number to the end of the file-name. For example if the file-name were set to "BACKUP" and the version number to 39, the file saved on disk would be under the name:

"BACKUP V50"... "BACKUP V51"...
"BACKUP V52"...

As you can see, OOPS! automatically increments the version number each time it saves a backup copy. In the unlikely event that the version number should exceed 9999, it is reset to zero. The **NUMBERMARK** command takes no arguments, so needs no error messages.

TIMEMARK

As an alternative to marking backup copies with a version number, you can add the current clock time to the filename instead. This allows you to pinpoint the exact version of a program you want, to the nearest minute. For instance, if you had entered **SETSAVING IN: FILENAME "TEST": TIMEMARK**, the backup copies saved would have been the names shown below, assuming that saving started at two o'clock in the afternoon.

"TEST 14:00"... "TEST 14:05"...
"TEST 14:20"...

This function of OOPS! is most useful when you are making frequent changes to a program. The **TIMEMARK** command gives no error messages.

UNMARKED

If you don't want the backup copies of your program to be marked with version number or time, you can enter the command **UNMARKED** to disable these facilities. From then on, backup copies will have the name given to them in the **FILENAME** command, with no suffix. However, there is a fallacy in this - You cannot have more than one file with the same name on a disk, so after the first backup copy has been saved, all subsequent ones will not result in a file being saved onto the disk. To overcome this problem, use the next action on the **REPLACING ON/OFF** command.

REPLACING ON/OFF

As more and more backup copies of a program are saved on a disk, so the amount of free space remaining gets smaller and smaller until the disk overflows. If you are working on a long program there may not be room for more than three copies of so on one disk. The answer to this problem and also to saving backups with no suffix is to erase the last backup copy before you save the new one. In this way the

amount of room left on a disk will only decrease by the amount you add to a program between saves. For instance if you had entered **VERSION IS: NUMBERMARK: FILENAME "OOPS": REPLACING ON**, the following process would take place:

Save "OOPS" V1"...Erase "OOPS-V1" and Save "OOPS" V2"...

As you can see, this saves a large amount of space on a disk, but also means that your files are not quite so secure because the last copy has to be erased, and it is also slower than just saving new copies. When the first copy of a program is saved with **Replacing On**, OOPS! erases that this is the first save, and so does not attempt to erase a non-existent previous file **REPLACING** gives only the **SYNTAX ERROR** message, if you omit **ON & OFF**.

SAVING ON/OFF

When you have decided what to call your backup copies, how to mark them and whether or not to erase the last one etc, you can enable the OOPS! saving function using **SAVING ON**. To switch it off again just use **SAVING OFF**. Once saving is switched on, when it is time to save a backup copy (as dictated using **SETSAVING**) the following happens:-

1. The line you are entering is erased.
2. The messages below are displayed and their corresponding actions carried out.
- ERASING "Last File-name" if **REPLACING ON**
Last backup copy is erased if **REPLACING ON**
SAVING "in New File-name"
3. Your BASIC program is saved.
4. The Computer returns to Direct mode.

As you can see from the above, each time a program is saved the line that you are entering at that time has to be erased. This may seem a bit inconvenient, but when you think about it, the more you can lose is 80 characters of program, whereas if you hadn't saved your program you could have lost all a kilobyte of it!

The saving function of OOPS! only waits from Basic Direct mode (When entering programs). If it is time to save a backup copy while a program is running, OOPS! simply waits for the next time and then tries again. In this way it does not interfere with the

running of the program.

SAVING gives the **SYNTAX ERROR** message if the **ON & OFF** argument is left out.

STORE (CTRL-X)

Sometimes, you may want to save extra backup copies of a program separately from the timed saving function. It is annoying to have to wait ten minutes (or whatever) for the next automatic backup, but you may still want the program to be saved with a version number or the time etc. To save a program at any time, using the marking etc., you can simply type **STORE**. This has the same effect as if it were time to save a backup copy automatically. Additionally, the **STORE** command can be used even when automatic saving has been disabled using **SAVING OFF**.

Again, for those who consider typing five characters a major undertaking, you can achieve the same effect as typing **STORE** by simply pressing **CTRL-X**. Yes, I would have liked to use **CTRL-S** too, but Commodore beat us to it, for stopping printing!

OOPS! Information Commands

As you can see, OOPS! has a large number of functions which, when you first use the system, can be more than a little confusing. To help you when you are just starting to use OOPS! I have included in the program five commands to give you information on all aspects of OOPS! from its commands to the current settings of the alarm or the automatic saving. I'll now discuss each of these in turn.

COMMANDS

OOPS! has thirty commands in all, and you may find it hard to remember all of them. If this is the situation, just type **COMMANDS**. This results in a heading and a list of all thirty commands as reference. You should then not find it too hard to remember which command does what.

CLINFO

Apart from the **COMMANDS** command, there are also four specialized information commands which cover the clock, alarm, events and saving functions of OOPS! The first of these is **CLINFO**.

On entering **CLINFO**, a list of information about the OOPS! clock is printed on-screen, which is as follows:-

TIME: HH:MM:SS - The current time of the clock.
DISPLAY: ON/OFF - Whether the clock display is on or off as set up by the **CLOCK ON/OFF** command.
STATUS: STARTED/STOPPED - Whether the clock is running or not, as dictated using the **START** and **STOP** commands.

ALINFO

The **ALINFO** command is similar to the **CLINFO** command, but is used to print information about the GOPSE alarm instead of the clock. It gives the information shown below.

TIME: HH:MM - The time at which the alarm is set to go off.
STATUS: ON/OFF - Whether the alarm is enabled or disabled using the **ALARM ON/OFF** command.
MESSAGE: "Warning" - The warning message to be displayed when the alarm time is reached, as defined using the **WARNING** command.

EVINFO

The **EVINFO** command displays a list of information about the current settings for each of GOPSE's eight events. All are listed under a set of headings which may be explained as follows:-

- The event number to which this line of information refers.
TIME - The time at which this event will occur.
STS - The status of this event - whether or not it is on or off as defined using the **ENABLE** and **DISABLE** commands.
MESSAGE - The event message assigned to this event using the **MESSAGE** command.

Each line shows the same information, but for event number as shown on the left of the line.

SVINFO

The last of the GOPSE information commands is the **SVINFO** command which is used to display the current settings for the automatic saving function in GOPSE. The list of information the command gives is shown below.

TIMING: MM - Period between subsequent saves in minutes as defined by the **SETSAVING** command.
STATUS: ON/OFF - Whether or not the automatic saving is on or off, as set up using **SAVING ON/OFF**.
MODE: UNMARKED/NUMBERED - The current method for marking backup copies.
TIMEMARK
REPLACING: ON/OFF - Whether or not the replacing or saving function is enabled by the **REPLACING** command.
VERSION: VVVV - The current version number set up using the **VERSION** command.
DEVICE: DD - The current saving device number, as set up using the **DEVICE** command.
FILENAME: "Filename" - The current save file-name defined with the **FILENAME** command.

Miscellaneous Commands

As well as the specific function commands and information commands, I have provided GOPSE with a few extra commands to make life a bit easier when using the system. This section deals with these three remaining commands, namely **WINDOW**, **RECOVER** and **RESET**.

WINDOW MDW:

You may have noticed that when the clock display is enabled, scrolling the screen causes the clock to blink in an annoying way. You may also have seen that some strange effects can be achieved using the Delete and Insert keys on the top two lines of the screen. It would be better if all work took place in an area of the screen not affected by the clock display or flashing messages, which would remedy the problems outlined above. The answer is the **WINDOW** command.

The **WINDOW** command is used to reduce the usable area of the screen to an area below the clock display so that it does not interfere with any work you're doing. It is the equivalent of moving the cursor to the corners of the area and pressing **ESC-F** and **ESC-S**. Two windows are available which are similar to the **ESC-N** screens, but with the top edge below the clock display. The **WINDOW** command takes only one argument - **WINDOW #** enables the **ESC-N** equivalent window, and **WINDOW 1** the **ESC-R** equivalent. These are best explained by trying them out yourself. The edge coordinates for the respective windows are as follows:

WINDOW 0 Top-Left Corner: (0,2)
 Bottom-Left Corner: (58,24)

WINDOW 1 Top-Left Corner: (1,3)
 Bottom-Left Corner: (58,25)

The **WINDOW** command gives the error messages shown below.

INVALID ERROR - You have entered the window number argument.

TYPE MISMATCH ERROR - You have mistakenly replaced the window number with a string.

ILLEGAL QUANTITY ERROR -

The window number you have specified is neither 0 or 1 as it should be.

RECOVER

In addition to the automatic backup facility of OOPS! to stop you losing valuable programs, OOPS! also has an **UNNEW**-type command **RECOVER**. If you enter a program and then erase it from memory using the **NEW** command, you can get it back by typing **RECOVER**. The command will of course not work after you pass the **RESET** button on your Plus4, doing this erases the OOPS! program code. Some of the time you may be able to get your program back in this situation by reloading the OOPS! program then typing **RECOVER**. You should not try to use the command when you've just switched your computer on, because it will give some funny results with no program data present.

The **RECOVER** command has no arguments so it yields an error message.

RESET

The final OOPS! command, **RESET**, is used to reset the OOPS! data to its default values. This turns the clock on, resets all the events and saving on, to the values that they have when you first use OOPS! These default values are setup as typical values such as you might frequently use. These default settings are shown here.

OOPS! Abbreviations

Like the normal Commodore Basic commands, OOPS! commands can be typed abbreviated to save on typing. When typed into a program abbreviated, they are converted into their full form on listing so are just as easy to understand as typing them in fully. A list of the OOPS! command abbreviations is provided here. All abbreviations take exactly the same syntax as the full command.

OOPS! Kickstart File Generator

You should now have sufficient knowledge to be able to use all of the OOPS! commands competently. I'll now discuss the OOPS! Kickstart File Generator program (Listing # 2).

Most of the time when using OOPS! you will find that you always use the same saving setup, or the same event times and messages, so it is a

Command	Arguments	Minimum Abbreviation
SETCLOCK	"HH:MM:SS"	S SHIFT-E
CLOCK	ON/OFF	C SHIFT-L
SETALARM	"HH:MM"	SET SHIFT-A
WARNING	"Message"	W SHIFT-A
ALARM	ON/OFF	A SHIFT-L
SETEVENT	EV, "HH:MM"	SET SHIFT-E
MESSAGE	EV, "Message"	M SHIFT-E
ENABLE	EV1, EV2	E SHIFT-N
DISABLE	EV1, EV2	D SHIFT-I
OK	-	OK
SETSAVING	SAV	SET SHIFT-S
DEVICE	DEV	D SHIFT-E
FILENAME	"Filename"	F SHIFT-I
VERSION	VER	V SHIFT-E
NUMBERMARK	-	N SHIFT-U
TIMEMARK	-	T SHIFT-I
UNMARKED	-	U SHIFT-N
REPLACING	ON/OFF	R SHIFT-E
SAVING	ON/OFF	S SHIFT-A
STORE	-	S SHIFT-T
CSTART	-	C SHIFT-A
CSTOP	-	CST SHIFT-O
RESET	-	RE SHIFT-S
RECOVER	-	RE SHIFT-C
WINDOW	0/1	W SHIFT-I
COMMANDS	-	C SHIFT-G
CLINFO	-	CL SHIFT-I
ALINFO	-	AL SHIFT-I
EVINFO	-	E SHIFT-V
SVINFO	-	S SHIFT-V

bit of a waste of time to keep on typing the list of **SETEVENT** commands or whatever to use these values every time you use OOPS! or every time you reset your Plus4. It would be much easier to just write a short "Kickstart" program to set up the OOPS! functions so that all you need to do whenever you use OOPS! is to load and **RUN** this program. This is of course possible, and you can write such a program like any other Basic program using the OOPS! commands, but as an alternative, I have included the OOPS! Kickstart File Generator which can write such a program itself - A program writing another program!

When you run the Kickstart File Generator (Hereafter referred to as **KICEGEN**), it will ask you a series of questions. When it has done this, it will create a Kickstart program and then save it to disk or cassette. In this section I'll discuss all the questions **KICEGEN** asks, with the expected replies shown within parentheses.

(1) **WINDOW (0/1)** Enter which of the windows you wish to start using OOPS! within: The **ESC-N** type window 0, or the **ESC-R** window 1.

(2) **CLOCK TIME (HH/MM/SS)** Enter the time you wish the OOPS! clock to be set to when you first start using OOPS! Note that when you are entering times into **KICEGEN**, you must separate the Hours, Minutes and Seconds with any character EXCEPT a colon. For example 10/09/00, 10-09-00 and 10_09_00 are all valid, but 10:09:00 will not work as the BASIC input routine cannot cope with it.

(3) **CLOCK DISPLAY (ON/OFF)** Enter whether or not you want the clock display switched ON or OFF when you start using OOPS!

(4) **DO YOU WANT TO SET THE ALARM? (YES/NO)** If you don't want to bother with setting the OOPS! alarm, just enter **NO** and go on to section (5). If you do want to set the alarm, follow the steps below.

(5) **ALARM TIME (HH/MM)** Enter the time you want to set the alarm to.

(6) **ALARM WARNING (Message)** Enter the message that you want to be displayed on-screen when the alarm time is reached.

(7) **ALARM STATUS (ON/OFF)**
Enter whether or not you want the alarm to be switched ON or OFF when you start using GOFPS!

(8) **DO YOU WANT TO SET ANY EVENTS? (YES/NO)** If you don't want any events set up when you start using GOFPS! enter NO and go to step (12). Otherwise, enter YES and follow the steps below.

(9) **EVENT X TIME (HH/MM)**
When you have answered that you do want to set some of the GOFPS! events, you will be asked a series of questions for each of the events 1-6. The prompts displayed are of the same form for each event, with X being the event number to which the query refers. E.g. "EVENT 3 TIME" or "EVENT 7 STATUS". The first entry, EVENT TIME is the time you want this particular event to be set to, but if you don't want to set this event, just press RETURN without entering anything and go on to the next event.

(10) **EVENT X MESSAGE (Message)**
Enter the message you wish to be displayed when the time for event X is reached.

(11) **EVENT X STATUS (ON/OFF)**
Enter whether or not you want event number X to be enabled (ON) or disabled (OFF).

(12) **DO YOU WANT TO SET THE SAVING? (YES/NO)** If you want to set up the automatic saving facility of GOFPS! enter YES and follow the steps below. If not, enter NO and skip to the end of this section.

(13) **SAVE TIMING (0-60)** Enter the period of time (in minutes) between subsequent automatic saves.

(14) **DEVICE NUMBER (0-11)** Enter the device number of the disk drive on which you wish GOFPS! to save its automatic backup copies.

(15) **FILENAME (Name)** Enter the filename under which you want backup copies of a program to be saved.

(16) **VERSION NUMBER (0-9999)**
Enter the version number that you wish the first backup copy to be marked with when VERSIONMARK is executed.

(17) **MARKING MODE (N,T,I)**
Enter the first character of the mode

by which you wish backup copies to be marked. This should be one of (N)umbermark, (T)ime mark or (I)ncrement.

(18) **REPLACING (ON/OFF)** Enter whether or not you wish the previous backup copy to be erased when a new backup copy is saved.

(19) **SAVING STATUS (ON/OFF)**
Enter whether or not you want the automatic saving function of GOFPS! to be switched ON or OFF when you first start using the system.

When you have answered all the questions the screen window will clear and the message "COMPILING KICKSTART FILE..." will be displayed. After this, program lines will gradually be displayed which set up all the GOFPS! functions as you have specified by answering KICKGEN's questions. When KICKGEN is displaying program lines, what it is doing is actually building a BASIC program into another area of memory from itself. When it has finished this process, KICKGEN displays the message "COMPILING COMPLETE - TAPE/DISK (T/D)?". If you want to save the Kickstart program to Tape, press "T" or if to Disk, press "D". KICKGEN then asks you to insert a disk or tape, and then when you press RETURN it will save the Kickstart program onto the tape or disk.

When KICKGEN has saved the Kickstart program, from that time you can set up your preferred GOFPS! functions simply by typing the following.

DLOAD "KICKSTART" If you are using a disk-drive.

LOAD "KICKSTART" If you are using a Datasette as a more secure media.

When the program has loaded, simply type RUN and hit RETURN. Of course, don't forget to load the GOFPS! main program before!

That completes our discussion of the GOFPS! commands and KICKGEN program. I'll now discuss the most important aspect of GOFPS! - Getting the thing started!

The GOFPS! system comes in two parts which are the GOFPS! Basic Loader (Listing # 1) and the GOFPS! Kickstart File Generator (Listing # 2). The Basic Loader is simply used to create the machine-code GOFPS! program, and is used as follows:

Enter Listing # 1 and RUN. The program will give messages to allow you to correct the data lines. Note that you should start the section of data with line 1000. To help you, you can use the line below to set up automatic line numbering, and function key 1 to produce the DATA command with one keypress:-

KEY 1, "DATA": AUTO 10

When you have entered Listing # 1 and corrected all your mistakes, the program will ask you whether you want to save the GOFPS! programs to disk or tape. Press "D" or "T" to choose which and then insert a datasette or tape and press RETURN to save the programs. The programs saved are: (a) BASIC Loader and (b) GOFPS! Machine-Code Program. When these programs have been saved, to use the GOFPS! system just reset your Plus/4 and type the following:

**For Tape Users: DLOAD "GOFPS!"
RETURN RUN RETURN**

**For Disk Users: DLOAD "GOFPS!"
RETURN RUN RETURN**

When you type RUN, the GOFPS! commands are enabled, a title screen displayed and the clock switched on. You can then write programs or use GOFPS! commands. If you use the RUN/STOP-RESET combination to escape from a machine-code program, typing X RETURN from TED-MON will get you back into Basic and re-enable the GOFPS! functions.

As mentioned before, the saving facility of GOFPS! cannot be used with a Datasette, but all the clock, alarm and event functions may be used with either disk or tape.

The KICKGEN program (Listing # 2) can be entered just like any other normal BASIC program. It can be used as shown below.

**For Tape Users: SAVE "GOFPS!"
KICKGEN" RETURN**

**For Disk Users: DSAVE "GOFPS!"
KICKGEN" RETURN**

The GOFPS! KICKGEN program may be used either with or without GOFPS! resident in memory.

Believe it or not, that concludes this article about GOFPS! Now then, I'll just format that disk... there we go... hang on, what does that disk-label say? "GOFPS! Master disk?... Oh ****!"

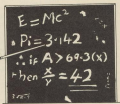
PLUS! PROGRAM

[illegible][illegible]

30000 D404 49, 48, 77, 48, 80, 14, 08, 0
 0, 0, 0000
 30001 D404 48, 47, 48, 54, 30, 82, 77, 8
 0, 0, 0000
 30002 D404 52, 47, 48, 77, 80, 83, 77, 8
 0, 0, 0000
 30003 D404 47, 40, 11, 80, 80, 80, 80, 0
 0, 0, 0000
 30004 D404 80, 80, 14, 80, 00, 70, 00, 0
 0, 0, 0000
 30005 D404 52, 04, 30, 74, 00, 40, 80, 1
 4, 0, 0000
 30006 D404 49, 49, 00, 00, 70, 00, 00, 0
 0, 0, 0000
 30007 D404 70, 04, 00, 00, 80, 00, 10, 0
 0, 0, 0000
 30008 D404 53, 60, 00, 00, 80, 10, 80, 0
 0, 0, 0000
 30009 D404 10, 80, 60, 80, 80, 40, 80, 0
 0, 0, 0000
 30010 D404 50, 01, 40, 00, 71, 40, 80, 0
 0, 0, 0000
 30011 D404 80, 83, 00, 14, 00, 00, 00, 0
 0, 0, 0000
 30012 D404 10, 80, 80, 80, 14, 00, 07, 0
 0, 0, 0000
 30013 D404 07, 40, 80, 14, 00, 10, 00, 0
 0, 0, 0000
 30014 D404 80, 00, 80, 80, 14, 00, 80, 1
 0, 0, 0000
 30015 D404 40, 00, 80, 80, 00, 00, 14, 0
 0, 0, 0000
 30016 D404 40, 00, 80, 80, 00, 70, 07, 0
 0, 0, 0000
 30017 D404 80, 00, 80, 80, 00, 70, 07, 0
 0, 0, 0000
 30018 D404 80, 00, 80, 80, 00, 00, 14, 0
 0, 0, 0000
 30019 D404 80, 00, 80, 80, 00, 00, 14, 0
 0, 0, 0000
 30020 D404 07, 00, 00, 00, 80, 14, 40, 0
 0, 0, 0000
 30021 D404 00, 00, 14, 80, 18, 00, 07, 0
 0, 0, 0000
 30022 D404 00, 00, 40, 80, 80, 14, 00, 0
 0, 0, 0000
 30023 D404 07, 40, 00, 14, 00, 80, 07, 0
 0, 0, 0000
 30024 D404 00, 40, 00, 80, 80, 00, 00, 1
 4, 0, 0000
 30025 D404 00, 00, 14, 80, 80, 14, 00, 0
 0, 0, 0000
 30026 D404 00, 00, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30027 D404 00, 00, 80, 70, 70, 40, 80, 0
 0, 0, 0000
 30028 D404 00, 00, 80, 70, 10, 40, 80, 0
 0, 0, 0000
 30029 D404 11, 00, 00, 80, 40, 80, 14, 0
 0, 0, 0000
 30030 D404 04, 00, 00, 80, 14, 40, 80, 1
 4, 0, 0000
 30031 D404 00, 70, 10, 40, 80, 14, 00, 0
 0, 0, 0000
 30032 D404 00, 80, 14, 80, 80, 14, 00, 0
 0, 0, 0000
 30033 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30034 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30035 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30036 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30037 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30038 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30039 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30040 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30041 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30042 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30043 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30044 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30045 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30046 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30047 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30048 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30049 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30050 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30051 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30052 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30053 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30054 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30055 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30056 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30057 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30058 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30059 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30060 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30061 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30062 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30063 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30064 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30065 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30066 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30067 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30068 D404 00, 80, 14, 00, 80, 00, 80, 0
 0, 0, 0000
 30069 D404 00, 80, 14

[illegible][illegible][illegible]

Logically Speaking



*Get your thinking caps on for this excursion into
Mathematics and Logic
By Pete Gerrard*

I would imagine that many of you have dabbled with the game of Chess at one time or another, or perhaps a simpler game of Draughts on the same board. The complexities of Chess can be baffling for the beginner, the rules of Draughts are never really understood, but nevertheless both of them remain fascinating games for expert and novice alike. The topic under discussion now revolves around the board rather than the pieces on it, so, if you have such a board, you may care to get it out and study it. If not, you will just have to imagine an eight by eight board; don't worry about black and white squares, just stick to the eight by eight board, and we can continue.

After a quick calculation you might say that there are 64 squares on the board, and indeed there are 64 squares that are just one square in width and height, but what about the squares that are two squares in width and height, or three squares, or even eight

squares? Of course, there is only one of the latter, but there are many more of the other sizes and all the intermediate ones. How many are there in total?

As if that were not puzzling enough, consider also that there are many, many more rectangles formed than there are squares. Rectangles can be two squares by one, or four squares by five, and there are far more of them than the more conventional squares.

So how many rectangles can be formed on a standard Chess board? and having discovered that one, what is the final total for the combined number of rectangles and squares?

Fortunately for us there are mathematical formulae for solving these particular riddles. However, if I were to simply tell you the answer for a square the size of a chessboard, that would not tell you the solution for a nine by nine board, or an even greater function by four squares. Hence it is here once more in the form of the accompanying program. This will allow you to solve the riddle of how many squares and rectangles there are for a whole range of squares from the simple two by two to the much more

complex twenty by twenty.

Of course, you could use even larger numbers, although the square itself would not be drawn on the screen and the listing would need amending accordingly.

First, let us consider the formulae. If we have a variable 'A' equal to the size of the square (and thus 'A' would equal 8 for our Chess board problem), then the combined total of squares and rectangles that can be formed is given by the formula $(A^2 + A) \cdot (A^2 + A) / 12$. This gives us the total number and the individual number of squares is given by the formula $(2 \cdot A^3 + A^2 + A) / 6$. From this we can of course work out the number of rectangles easily enough, but just for the sake of completeness, the formula that itself gives us the number of rectangles not including squares is $(3 \cdot A^3 + A^2 + 2 \cdot A^2 + 2 \cdot A) / 12$. A complicated one!

That brings us to the program listing, and a simple enough one it is too once we know the relevant formulae. Lines 10 and 20 serve to identify it, while lines 30 and 40 select our colour scheme and clear the screen for neatness. Lines 50 to 60 then allow the

user to input a number in the range 1 to 30, and could be amended if bigger squares were required, although you would then be advised not to attempt drawing them and skip straight to line 140.

Line 70 clears the screen again, lines 80 to 130 define and draw the square in its entirety, before we get to the actual calculations. After a brief statement in line 140, line 150 then works out the total number of squares and rectangles using the formula given earlier. Line 160 calculates the number of squares and the number of rectangles including squares. That's the formulae do actually give us the correct answer can soon be verified by drawing a trivial square two by two, the instance, which is easily calculated by the eye.

The answers are displayed by lines 170 to 190, and line 200 rounds everything off and prevents the program falling through to line 210 and producing a RETURN WITHOUT GOSUB error.

That is the program, these are the formulae, and an interesting little mathematical diversion they form as well. However, an article of this nature would not be complete without giving you some work to do as well. As we have seen, there are formulae for squares, but what about triangles? Imagine an equilateral triangle that is divided into eight smaller triangles along its base, right along, its two remaining sides, and the interior filled in with triangles of the same size. Rather like an eight by eight chess board, we might be said to have an eight by eight by eight triangle. Remembering that triangles can be formed upside down as well as the right way up, what is the total number of triangles contained within the big triangle, saying that the big triangle is a white side one to the total, just as the eight chess board adds one to the number of squares. Easy if you know the formula, but if you don't....!!

Switching The Bases

As human beings we are accustomed to reasoning in a numerical system that uses ten as its base, hence we arrive at what is called a decimal system. This is not particularly convenient for computers which are ideally suited to the so-called binary system.

As far as computers are concerned an electronic circuit can be either

turned on or off, it only has two possible values associated with it, and, if we give those values the numerical associations of zero and one, we have the basis for the binary system.

Binary and decimal do not mix happily though, and sometimes counting systems come into the reckoning. This is illustrated here, dealing with the number 16 as its base. Unfortunately the Arabic numerals from which our own derive only ascend as far as 9, using the symbols 0, 1, 2, and so on. Rather than inventing new symbols, hexadecimal uses letters of the alphabet. These letters are A, B, C, D, E and F, and this gives us hexadecimal numbers of the form 13FE for example.

In order to convert numbers from one numerical system to another, we need only know the base that is being used. For instance, the decimal number 1234 is easily translated as 4 plus 3 times 10 plus 2 times 10 to the power 2 plus 1 times 10 to the power of 3. The further left we go in the number, the greater the power of ten that is being used. To take our rather hexadecimal example of 12FE, knowing that E is equivalent to ten decimal, number 16 and F is equivalent to the decimal number 15, we arrive at 14 plus 13 times 16 plus 2 times 16 to the power of 2 plus 1 times 16 to the power of 3, or 1790 in our more familiar decimal system.

Thus the basis for all counting systems rests in the base being used, and by multiplying that base number raised to the appropriate power by the number concerned, we can easily convert from one numerical base to another. To give another example, let us assume a numerical system that uses the base 7. An unusual choice, perhaps, before inventing one ourselves. If we look at a number of 1234 which was expressed using this base of 7, to convert it back into our own decimal system we have 4 plus 3 times 7 plus 2 times 7 to power 2 plus 1 times 7 to power 3. This equates to a decimal number of 464.

Now we arrive at the mathematical diversion. To begin with, we will be using a base of 8, a base of 5, and a base of 7. In other words, three different bases. What we want to find is a solution to the following equation, assuming the variable 'A' to be the number we are looking for: $A \text{ (base 8) minus } A \text{ (base 5) equals } A \text{ (base 7)}$. The same number, but using a

different numerical base for each time that it is expressed. What is more, we want to find a four digit number that will obey the equation, and, to complicate matters still further, we would like to find a number which has its second and third digits equal. A number like 1233, for example, although that is not the answer.

Many such problems can be devised, and I shall leave you with one to ponder on at the end of this article. For now, let us consider the problem as it stands and the program below that enables us to solve it. With the rules of the problem clearly expressed, and the method of translating numbers from one base to another already defined, the listing should pose no problems.

Line 10 serves to identify the program and its author before line 20 sets our string variable, while line 30 clears the screen. The loop in line 40 'loops' up to cover all possible four digit numbers, and with lines 50 and 60 we set up a string variable so that we can check for the presence of two equal digits in the second and third positions within these four digit numbers.

If those digits are not the same then line 70 sends us straight to line 100 to continue the search, ultimately concluding with that line if no such number is found. However, if the second and third digits are the same then we go on to lines 110, 130, 150 and 170 in order to convert that number to base 8, 5, 7 and 10 respectively, before going to line 190 and displaying the number on the screen. So inform the user of the program being made.

If the equation now holds true then line 90 tells us of the relevant number before halting the program with an END statement, otherwise we carry on with the loop in line 40.

The first number that conforms with our equation is the decimal number 1332. That is, 1332 in base 8 minus 1332 in base 5 equals 1332 in base 7. The decimal number 1666 also obeys the rules.

That is a simple example of the kind of problem that can be set when using a mixture of bases for counting purposes. Now consider a different sort of problem using the equation $A \text{ (base 4) equals } A \text{ (base 3) plus } A \text{ (base 5)}$. If the numeric variable 'A' is representing the decimal number 1771, what are the values of a_3 and a_5 in other words, what bases are we using?



PROGRAM: BASICS

```

80 10 REM BASICS BY PETE GERRARD

83 20 X=5000:10:POKE X,2:POKE X
+1,0:PRINT CHR$(150)
89 30 PRINT CHR$(147)
98 40 FOR I=1000 TO 9999
99 50 A=1
99 60 AS=MID$(STR$(A),2)
99 70 IF MID$(AS,2,1)<>MID$(AS,
3,1) THEN 100
99 80 GOSUB 110:GOSUB 130:GOSUB
150:GOSUB 170:GOSUB 190
99 90 IF AS=BS=CS THEN GOSUB 19
0:PRINT:PRINT BS;" IS ONE PO
SSIBLE ANSWER."END
99 100 NEXT I:PRINT "NO NUMBER
FOUND."END

99 74 110 REM BASE 6
99 87 120 A1=0:FOR J=1 TO 4:X$=MID
$(AS,J,1):X=VAL(X$):A1=A1*X*
6^(4-J):NEXT A1:RETURN
99 93 130 REM BASE 5
99 100 140 A1=0:FOR J=1 TO 4:X$=MID
$(AS,J,1):X=VAL(X$):A1=A1*X*
5^(4-J):NEXT A1:RETURN
99 106 150 REM BASE 7
99 113 160 A1=0:FOR J=1 TO 4:X$=MID
$(AS,J,1):X=VAL(X$):A1=A1*X*
7^(4-J):NEXT A1:RETURN
99 119 170 REM BASE 10
99 126 180 A1=0:FOR J=1 TO 4:X$=MID
$(AS,J,1):X=VAL(X$):A1=A1*X*
10^(4-J):NEXT A1:RETURN
99 132 190 REM DISPLAY NUMBERS
99 139 200 PRINT CHR$(147);
99 146 210 PRINT "BASE 6 =",A1
99 153 220 PRINT "BASE 5 =",A1
99 160 230 PRINT "BASE 7 =",A1
99 167 240 PRINT "BASE 10 =",A1:BS=
MID$(STR$(A1),2)
99 174 250 RETURN

```

PROGRAM: SQUARES

```

83 10 REM SQUARES BY PETE GERRARD
83 20
83 30 REM FOR THE COMMODORE 64
83 40 30 X=5000:10:PRINT CHR$(147)
+POKE X,2:POKE X+1,0:PRINT C
HR$(150)
83 40 PRINT CHR$(147)
83 50 INPUT "SIZE OF SQUARE (E.
S. S FOR 8 BY 8)SPACES: ",A
83 60 IF A=2 OR A=20 THEN 40
83 70 PRINT CHR$(147)
83 80 A=A-1:BS="":BS="":CS=""
83 90 FOR I=1 TO A:BS=BS+CHR$(
78):NEXT:AS=CHR$(179):BS=BS+
CHR$(179)
83 94 100 FOR I=1 TO A:BS=BS+CHR$(
179):NEXT:BS=BS+CHR$(179):BS=BS+
CHR$(179)
83 99 110 FOR I=1 TO A:CS=CS+CHR$(
177):NEXT:CS=CS+CHR$(179):CS=CS+
CHR$(179)
83 106 120 REM STRINGS DEFINED
83 113 130 PRINT AS:FOR I=1 TO A:PR
INT BS:NEXT:PRINT CS
83 119 140 X=BS\Y=0:GOSUB 210:PRINT
" SQUARES & SQUARES"
83 126 150 A=A+1:Y=Y+3:YR=(A+A+A)
*(A+A+A)/4
83 132 160 BS=(2*A*A*A-2*A*A-A)/5:Y
R=(2*A*A*A*A-2*A*A*A-2*A*A-A)
*(A)/12
83 139 170 GOSUB 210:PRINT "TOTAL S
QUARES & Y-Y-1:GOSUB 210:PR
INT "RECTANGLES =",YR
83 146 180 Y=Y+2:GOSUB 210:PRINT "S
QUARES:SPCH="AS
83 153 190 Y=Y+2:GOSUB 210:PRINT "R
ECTANGLES =",YR
83 160 200 S=Y-Y-20:GOSUB 210:END
83 167 210 PRINT CHR$(147):FOR I=1
TO Y:PRINT:NEXT:FOR I=1 TO X
:PRINT CHR$(88):NEXT:RETURN

```


Binders

Organise and protect your disk with Commodore Disk User disk binders and data disks.

Why not keep your Commodore Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder comes complete with 10 disk sleeves to organise and protect your program disks. Why not buy a disk binder to house all of your data disks? We can even supply Commodore Disk User data disks. The Commodore Disk User logo immediately identifies your disks and there's room to title them and document the disks details. Send for your disks and binders now!

Prices are as follows:

Commodore Disk User Binder £4.15, including

10 sleeves. Order code **BDYU1**

Commodore Disk User Binder with 10 sleeves and

10 disks, £5.15 Order code **BDYU2**

10 sleeves for insertion in binder, £1.50. Order code

BDSD1

20 sleeves for inclusion in binder, £1.75. Order code

BDSD2

10 Commodore Disk User data disks, £5.95. Order

code **BDGD10**

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST. Please allow 28 days for delivery.



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			
			TOTAL



NEXT MONTH

NIGHTBREED

Clive Barker spills his guts in an exclusive interview about his new film and the game to follow.

LIFESTYLES 2000

On the spot report.

DESERT ISLAND DISKS

Tony Crowther is the first to contribute to our new regular spot.

Plus Rule Playing Games, Play By Mail, Software in the Office...

Reasoning On The 128

*We continue with our expert system for the 128
computer*

By Paul Schofield

In the last installment of this series we introduced you to the idea of a simple expert system, from now on, we will look at the more sophisticated knowledge base which I introduced in that article. This system comprises of three programmes:

EDR - the source program editor
KBDEN - the knowledge base compiler
INQUIRE - the inference engine

Why three programs? The most important consideration is space available for data, which is increased by keeping the programs small. Speed of operation is also improved, as once the application has been debugged, it is only necessary to use the *INQUIRE* program which reads in the data in a compressed and preprocessed format. In this series, we will be concerned mainly with the editor, but first a quick overview of the whole system.

Defining a knowledge base

To create the knowledge base for the reason program, we first processed all the data that we had on the subject in question and arranged it into a tree structure. In the case of the very simple chemical analysis example, we ended up with a nice simple symmetric tree, which therefore took an efficient path to any of the solutions. If we were to extend this, we would have to start using tests to identify particular chemicals and so the tree would no longer be symmetric. This is similar to the car maintenance example, where

it was suggested that the first questions might be:

Is fuel tank empty?
Is battery in flat?

In this case, we pick these as starting questions, because they are easy to check and the produce a simple solution.

In terms of our expert system however, they represent a fundamentally different approach. We pick a solution to the problem, namely, just petrol in fuel tank and ask a set of questions to support it. If the answers are affirmative we have an answer, otherwise we take the next possible solution and test that.

This is the approach that our new knowledge base system will adopt. It is probably the most widely used in this type of application and has a number of advantages. There are, however, attendant disadvantages, and so the system must endeavour to take account of these.

Incomplete data

One of the first rules one learns as a programmer is, garbage in means garbage out, and this is particularly true of this type of expert system. To define your knowledge base, it is necessary to define your targets (the possible solution) and the conditions which must be satisfied for a particular target to be the solution. The order in which you define the targets also implicitly defines the order in which they are evaluated. This is good, because at the outset you can start by just entering conditions, which you

know to be true about a target. The disadvantage is easily shown by an example. Suppose you write a knowledge base for animal identification. The first target is cat and the conditions are:

Has four legs?
Has a tail?

In use, the knowledge base will tell you that a wide variety of animals are cats. The real message is 'according to the supplied criteria a possible solution is a cat'. For the expert system to be useful it's not sufficient for it simply to stop at the first solution, it must provide to additional options:

Why - A justification of the solution.
Also - Continue to look for other solutions.

With the aid of these tools it is possible to identify sections of the knowledge base, which are incomplete and identify suitable conditions to apply. We will consider such problems in a little more detail when we consider the source definition language and the inference engine, but this should suffice as a general introduction to the system.

EDR - The source file editor

The obvious question is why not use the Basic editor? One reason of course is that the Basic editor produces colonised program files, but in general it is more convenient to use either a general purpose text editor or one specific to the particular application. EDR is specific to this application, but

would not require much modification to make a general purpose text editor. To run the editor, simply enter: `run "EDR"`. The editor initializes various variables and then displays the main options menu. Most of these are self explanatory, so we will just consider option 1, `create/edit RAL`, source file. When this option is selected, a help bar is displayed at the top of the screen showing the functions assigned to the function keys. The remainder of the screen is either blank (`create`) or displays the start of the currently loaded file. The files produced are sequential files, so no line numbers are needed, just enter the text you require. The following special function keys are also supported:

```
CURSOR : Move up/down/left/
KEYS    right one character
DEL      : Delete character left of
          cursor
INSRT    : Insert blank character at
          cursor
F1       : Insert blank line
F2       : Delete current line
RETURN   : Move cursor to start of
          next line
F6       : Delete remainder of line
          to right of cursor
F4       : Copy line to paste buffer
          and delete it
F3       : Paste saved line at cur
F2       : Move cur to top of file
F7       : Move cur to end of file
HOME     : Move cur to top left of
          screen
CTRL/U   : Move up 20 lines or to
          top of file
CTRL/D   : Move down 20 lines or
          to end of file
F8       : exit from edit mode
```

One automatic function of the editor is that when the bottom of the screen is reached, the text is scrolled up 10 lines.

Modifying The Editor

The functions supported by the editor are all that you are likely to need for entering knowledge base definitions. For more general use a few modifications are likely to be needed. New edit functions can be added very easily. Add a text for the key in question between line 10360 and line 10560 and insert the corresponding subroutines somewhere between line 2090 and line 4090.

Two additional modifications may also be required:

1) Currently the number of lines of text is written at the start of the disk files. If this is considered undesirable, the disk read/write routines must be changed to detect eof.

2) Commodore Basic has trouble with commas when writing text to disk files unless special precautions are taken. As the knowledge base definition language does not use commas, this has not been done.

We shall now look at the definition language RAL and the compiler program KNOEN.

Having experimented with the REASON program, and typing in the EDR Editor, We will now look at the knowledge base definition. Later, we will look at the compiler program, KNOEN. First of all though, we will look at the definition language. As these 3 programs together represent the package Reason Version 1, I have called the language, Reason Application Language (RAL).

Before going into details though, a quick look at some of the objectives. Although KNOEN is technically a compiler, I prefer to think of it as a simple preprocessor. It does not generate machine code, but simply takes a description of data structures in a readable format and translates them into a set of assign variables for driving an inference engine. It was also considered important, that the compiler should be able to output the processed knowledge base on a different disk to the source file. This is administratively very convenient, but, if tedious disk swapping is to be avoided, requires that the entire source file is read before attempting to write the output file. This in turn dictates that large string storage requirements are provided and in order not to unduly restrict the size of applications the language becomes a little cryptic in places.

It is possible to change this approach, but I would recommend trying the existing system first, it's not really that difficult.

Structure of RAL

RAL is a block structured language like Pascal or C, but the syntax of the individual blocks is a bit more rigid.

Outside any of the blocks comment lines are permitted, so the basic structure of a RAL program is:

```
comments
RAL BLOCK
comments
:
comments
RAL BLOCK
comments
```

There are 3 types of RAL block-

HELP - Initial instructions displayed to user of base.
TARGETS - Define the set of target solutions.
CONDITIONS - Define the set of questions the user is asked.
RELATIONS - Define set of rules for evaluating targets.
IMPLIES - Defines how additional conditions may be inferred from a users response.

The structure of each block is as follows:

```
BLOCK-HEADER
Block comments
END [comments]
```

The square brackets [] indicate optional elements.

Comments

As mentioned previously, comments may be included anywhere outside a block, they have the format:

```
[any text string]
```

It is important that the '`'`' is the first character on comment lines and these are simply ignored by the compiler. It would not be difficult to permit comments within blocks, but it would greatly slow down the compiler and is not very useful.

Help Section

The help section is optional, but should normally be included in all RAL definitions. It simply defines a number of lines of explanatory text that will be displayed to the user before any questions are asked. It may typically be used to display a title, take credit for creating the knowledge base and providing any necessary background information. It is not necessary to include any information on how to respond to reason questions as this is automatically output by the INQUIRE program. The format of the help section is:

HELP *integer*
help-text-lines
END [**HELP**]

where:

integer defines the number of lines of help information following.

help-text-lines are the help information in the format to be displayed.

Targets Section

The targets section defines the text strings describing the target solutions and is obviously mandatory. The format of the targets section is:

TARGETS *integer*
target-line
END [**TARGETS**]

where:

integer defines the number of entries to be reserved for targets.

and *target-line* have the format:

[*index*] *target-text*

where:

index is an optional target number reference (1,2,3 etc.)

target-text is the text that will be displayed to describe the solution.

Note

The purpose of *integer* is to define internal storage requirements and so must be the same as or greater than the number of following targets. The *index* is purely a comment, a bit like a Basic line number. It is used in later sections to identify specific targets and is therefore useful in large knowledge bases. The first index is always 1 and they are allocated consecutively.

Conditions Section

The conditions section defines the questions which will be asked to the user and is obviously mandatory. The format of the conditions section is:

CONDITIONS *integer*
condition-line
END [**CONDITIONS**]

where:

integer defines the number of entries to be reserved for conditions.

condition-line have the format:

[*index*] *condition-text*

where:

index is an optional condition number

reference (1,2,3 etc.)

condition-text is the text of the questions displayed to the user.

integer and *index* are treated as for targets.

Relations Section

The relations section defines the rules for evaluating targets according to the answers to the condition questions. The results of the various conditions may be either:

irrelevant

(must be true for target to be a solution) or *relevant* (must be false for target to be a solution).

By default, all conditions are considered irrelevant unless mentioned explicitly within a rule in the relations section. The format of the relations section is:

RELATIONS
relation-rule
END [**RELATIONS**]

where:

relation-rule have the format: *target-index operator condition-index-list*

where:

operator is '+' following conditions must be true, or '-' following conditions must be false.

condition-index-list is a number of condition indices separated by spaces.

Implies Section

The implies section is optional. It defines the results of further conditions, which may be determined from the result of the current condition. The object of this section is to avoid asking questions whose results should already be known and to a lesser extent, avoid investigating impossible targets as quickly as possible. For example, if during the investigation of one target the question is asked "IS OBJECT BLUE?" and the answer is true, it is quite obvious that a later question "IS OBJECT RED?" must have the answer false. This implies section allows upto three additional results to be implied for both a true or false answer to a particular condition. Its format is:

IMPLIES
implication-rule
END [**IMPLIES**]

where the format of implication rules is:

result-condition *implied-condition*
IMPLIED-CONDITION [**IMPLIED-CONDITION**]

and the format of both *result-condition* and *implied condition* is:

'</' *condition-index*
 As with relation rules + = true and - = false.

If that sounds a little confusing, then have a look at the example program in figure 1. Yes we are back with our mixed up chemical bottles again! With only 3 targets and 4 conditions this is very simple and it is easy to see how the relations and implies sections are used.

It is always best to work through the targets in the order they are defined as this is also the order in which they will be tested. In this case we check firstly for a strong acid. For this to be the case, blue litmus must be turned strongly red. This is a rather subjective question, so the simpler condition blue Litmus Turned Red is considered first.

In the relations section, we use rules defined, the first saying that conditions 1 and 2 must be true and a second requiring conditions 3 and 4 to be false. It is quite obvious in fact that if 1 and 2 are true that 3 and 4 will be false, but this is defined in the implies section instead. The reason for doing it this way is that at some time the future it could be decided to reorganise to test targets in a different order. This becomes difficult if the rules of the relations section are incomplete.

KBGEN program

The structure of this program is quite simple. The main program reads in the RAL source file line by line and looks for the sections in the expected order. Within each section the lines are validated according to the defined syntax. Any errors encountered are reported and if the error is fatal the processing aborts. Standard subroutines are used for skipping spaces and finding and extracting various elements in the source lines. It should not be that difficult to make changes to the program if desired.

The most obvious change to consider is to open the output file on the same disk at the start of the program. As the format of the help, targets and condition sections is very simple, these can be written directly to the output file after being validated so the large string arrays are avoided. This saving


```

1000 REM *****
1010 REM FROM PAGE DOWN CTD
1020 EP=0:LF=0:FL=0:CL=0:LL=0:AL=0
1030 PRINT SQR: IF LF=0 THEN LF=1
1040 IF LF=1 THEN CL=CL+1:IF CL=16 THEN CL=0
1050 IF CL=1 THEN LL=LL+1:IF LL=16 THEN LL=0
1060 IF LL=1 THEN AL=AL+1:IF AL=16 THEN AL=0
1070 GOTO 1000
1080 REM *****
1090 REM INSERT NEW FUNCTION SUBS
1100 REM HERE
1110 REM GENERAL DISK ACCESS SUB
1120 REM *****
1130 REM "CURRENT DISK IN" =
1140 REM DISK DRIVE NO. (0-3)
1150 REM "CURRENT DATA (Disk A) NO" =
1160 REM DISK DRIVE NO. (0-3)
1170 REM "CURRENT DATA (Disk A) NO" =
1180 REM DISK DRIVE NO. (0-3)
1190 REM "CURRENT DATA (Disk A) NO" =
1200 REM DISK DRIVE NO. (0-3)
1210 REM "CURRENT DATA (Disk A) NO" =
1220 REM DISK DRIVE NO. (0-3)
1230 REM "CURRENT DATA (Disk A) NO" =
1240 REM DISK DRIVE NO. (0-3)
1250 REM "CURRENT DATA (Disk A) NO" =
1260 REM DISK DRIVE NO. (0-3)
1270 REM "CURRENT DATA (Disk A) NO" =
1280 REM DISK DRIVE NO. (0-3)
1290 REM "CURRENT DATA (Disk A) NO" =
1300 REM DISK DRIVE NO. (0-3)
1310 REM "CURRENT DATA (Disk A) NO" =
1320 REM DISK DRIVE NO. (0-3)
1330 REM "CURRENT DATA (Disk A) NO" =
1340 REM DISK DRIVE NO. (0-3)
1350 REM "CURRENT DATA (Disk A) NO" =
1360 REM DISK DRIVE NO. (0-3)
1370 REM "CURRENT DATA (Disk A) NO" =
1380 REM DISK DRIVE NO. (0-3)
1390 REM "CURRENT DATA (Disk A) NO" =
1400 REM DISK DRIVE NO. (0-3)
1410 REM "CURRENT DATA (Disk A) NO" =
1420 REM DISK DRIVE NO. (0-3)
1430 REM "CURRENT DATA (Disk A) NO" =
1440 REM DISK DRIVE NO. (0-3)
1450 REM "CURRENT DATA (Disk A) NO" =
1460 REM DISK DRIVE NO. (0-3)
1470 REM "CURRENT DATA (Disk A) NO" =
1480 REM DISK DRIVE NO. (0-3)
1490 REM "CURRENT DATA (Disk A) NO" =
1500 REM DISK DRIVE NO. (0-3)
1510 REM "CURRENT DATA (Disk A) NO" =
1520 REM DISK DRIVE NO. (0-3)
1530 REM "CURRENT DATA (Disk A) NO" =
1540 REM DISK DRIVE NO. (0-3)
1550 REM "CURRENT DATA (Disk A) NO" =
1560 REM DISK DRIVE NO. (0-3)
1570 REM "CURRENT DATA (Disk A) NO" =
1580 REM DISK DRIVE NO. (0-3)
1590 REM "CURRENT DATA (Disk A) NO" =
1600 REM DISK DRIVE NO. (0-3)
1610 REM "CURRENT DATA (Disk A) NO" =
1620 REM DISK DRIVE NO. (0-3)
1630 REM "CURRENT DATA (Disk A) NO" =
1640 REM DISK DRIVE NO. (0-3)
1650 REM "CURRENT DATA (Disk A) NO" =
1660 REM DISK DRIVE NO. (0-3)
1670 REM "CURRENT DATA (Disk A) NO" =
1680 REM DISK DRIVE NO. (0-3)
1690 REM "CURRENT DATA (Disk A) NO" =
1700 REM DISK DRIVE NO. (0-3)
1710 REM "CURRENT DATA (Disk A) NO" =
1720 REM DISK DRIVE NO. (0-3)
1730 REM "CURRENT DATA (Disk A) NO" =
1740 REM DISK DRIVE NO. (0-3)
1750 REM "CURRENT DATA (Disk A) NO" =
1760 REM DISK DRIVE NO. (0-3)
1770 REM "CURRENT DATA (Disk A) NO" =
1780 REM DISK DRIVE NO. (0-3)
1790 REM "CURRENT DATA (Disk A) NO" =
1800 REM DISK DRIVE NO. (0-3)
1810 REM "CURRENT DATA (Disk A) NO" =
1820 REM DISK DRIVE NO. (0-3)
1830 REM "CURRENT DATA (Disk A) NO" =
1840 REM DISK DRIVE NO. (0-3)
1850 REM "CURRENT DATA (Disk A) NO" =
1860 REM DISK DRIVE NO. (0-3)
1870 REM "CURRENT DATA (Disk A) NO" =
1880 REM DISK DRIVE NO. (0-3)
1890 REM "CURRENT DATA (Disk A) NO" =
1900 REM DISK DRIVE NO. (0-3)
1910 REM "CURRENT DATA (Disk A) NO" =
1920 REM DISK DRIVE NO. (0-3)
1930 REM "CURRENT DATA (Disk A) NO" =
1940 REM DISK DRIVE NO. (0-3)
1950 REM "CURRENT DATA (Disk A) NO" =
1960 REM DISK DRIVE NO. (0-3)
1970 REM "CURRENT DATA (Disk A) NO" =
1980 REM DISK DRIVE NO. (0-3)
1990 REM "CURRENT DATA (Disk A) NO" =
2000 REM DISK DRIVE NO. (0-3)

```


[illegible]

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 395–402

[illegible][illegible]

```

14700 PRINT "END PROGRAM ABORTED."
14710 END
14720 DEF FN@ND TARGET TESTS
14730 IF=0 THEN GOTO 14740 ELSE GOTO 14750
14740 PRINT "REMOVING TARGETS SECTION"
14750 GOTO 14760
14760 IF=0 THEN GOTO 14770 ELSE GOTO 14780
14770 PRINT "END"
14780 IF=0 THEN GOTO 14790 ELSE GOTO 14800
14790 PRINT "END"
14800 IF=0 THEN GOTO 14810 ELSE GOTO 14820
14810 PRINT "END"
14820 IF=0 THEN GOTO 14830 ELSE GOTO 14840
14830 PRINT "END"
14840 IF=0 THEN GOTO 14850 ELSE GOTO 14860
14850 PRINT "END"
14860 IF=0 THEN GOTO 14870 ELSE GOTO 14880
14870 PRINT "END"
14880 IF=0 THEN GOTO 14890 ELSE GOTO 14900
14890 PRINT "END"
14900 IF=0 THEN GOTO 14910 ELSE GOTO 14920
14910 PRINT "END"
14920 IF=0 THEN GOTO 14930 ELSE GOTO 14940
14930 PRINT "END"
14940 IF=0 THEN GOTO 14950 ELSE GOTO 14960
14950 PRINT "END"
14960 IF=0 THEN GOTO 14970 ELSE GOTO 14980
14970 PRINT "END"
14980 IF=0 THEN GOTO 14990 ELSE GOTO 15000
14990 PRINT "END"
15000 IF=0 THEN GOTO 15010 ELSE GOTO 15020
15010 PRINT "END"
15020 IF=0 THEN GOTO 15030 ELSE GOTO 15040
15030 PRINT "END"
15040 IF=0 THEN GOTO 15050 ELSE GOTO 15060
15050 PRINT "END"
15060 IF=0 THEN GOTO 15070 ELSE GOTO 15080
15070 PRINT "END"
15080 IF=0 THEN GOTO 15090 ELSE GOTO 15100
15090 PRINT "END"
15100 IF=0 THEN GOTO 15110 ELSE GOTO 15120
15110 PRINT "END"
15120 IF=0 THEN GOTO 15130 ELSE GOTO 15140
15130 PRINT "END"
15140 IF=0 THEN GOTO 15150 ELSE GOTO 15160
15150 PRINT "END"
15160 IF=0 THEN GOTO 15170 ELSE GOTO 15180
15170 PRINT "END"
15180 IF=0 THEN GOTO 15190 ELSE GOTO 15200
15190 PRINT "END"
15200 IF=0 THEN GOTO 15210 ELSE GOTO 15220
15210 PRINT "END"
15220 IF=0 THEN GOTO 15230 ELSE GOTO 15240
15230 PRINT "END"
15240 IF=0 THEN GOTO 15250 ELSE GOTO 15260
15250 PRINT "END"
15260 IF=0 THEN GOTO 15270 ELSE GOTO 15280
15270 PRINT "END"
15280 IF=0 THEN GOTO 15290 ELSE GOTO 15300
15290 PRINT "END"
15300 IF=0 THEN GOTO 15310 ELSE GOTO 15320
15310 PRINT "END"
15320 IF=0 THEN GOTO 15330 ELSE GOTO 15340
15330 PRINT "END"
15340 IF=0 THEN GOTO 15350 ELSE GOTO 15360
15350 PRINT "END"
15360 IF=0 THEN GOTO 15370 ELSE GOTO 15380
15370 PRINT "END"
15380 IF=0 THEN GOTO 15390 ELSE GOTO 15400
15390 PRINT "END"
15400 IF=0 THEN GOTO 15410 ELSE GOTO 15420
15410 PRINT "END"
15420 IF=0 THEN GOTO 15430 ELSE GOTO 15440
15430 PRINT "END"
15440 IF=0 THEN GOTO 15450 ELSE GOTO 15460
15450 PRINT "END"
15460 IF=0 THEN GOTO 15470 ELSE GOTO 15480
15470 PRINT "END"
15480 IF=0 THEN GOTO 15490 ELSE GOTO 15500
15490 PRINT "END"
15500 IF=0 THEN GOTO 15510 ELSE GOTO 15520
15510 PRINT "END"
15520 IF=0 THEN GOTO 15530 ELSE GOTO 15540
15530 PRINT "END"
15540 IF=0 THEN GOTO 15550 ELSE GOTO 15560
15550 PRINT "END"
15560 IF=0 THEN GOTO 15570 ELSE GOTO 15580
15570 PRINT "END"
15580 IF=0 THEN GOTO 15590 ELSE GOTO 15600
15590 PRINT "END"
15600 IF=0 THEN GOTO 15610 ELSE GOTO 15620
15610 PRINT "END"
15620 IF=0 THEN GOTO 15630 ELSE GOTO 15640
15630 PRINT "END"
15640 IF=0 THEN GOTO 15650 ELSE GOTO 15660
15650 PRINT "END"
15660 IF=0 THEN GOTO 15670 ELSE GOTO 15680
15670 PRINT "END"
15680 IF=0 THEN GOTO 15690 ELSE GOTO 15700
15690 PRINT "END"
15700 IF=0 THEN GOTO 15710 ELSE GOTO 15720
15710 PRINT "END"
15720 IF=0 THEN GOTO 15730 ELSE GOTO 15740
15730 PRINT "END"
15740 IF=0 THEN GOTO 15750 ELSE GOTO 15760
15750 PRINT "END"
15760 IF=0 THEN GOTO 15770 ELSE GOTO 15780
15770 PRINT "END"
15780 IF=0 THEN GOTO 15790 ELSE GOTO 15800
15790 PRINT "END"
15800 IF=0 THEN GOTO 15810 ELSE GOTO 15820
15810 PRINT "END"
15820 IF=0 THEN GOTO 15830 ELSE GOTO 15840
15830 PRINT "END"
15840 IF=0 THEN GOTO 15850 ELSE GOTO 15860
15850 PRINT "END"
15860 IF=0 THEN GOTO 15870 ELSE GOTO 15880
15870 PRINT "END"
15880 IF=0 THEN GOTO 15890 ELSE GOTO 15900
15890 PRINT "END"
15900 IF=0 THEN GOTO 15910 ELSE GOTO 15920
15910 PRINT "END"
15920 IF=0 THEN GOTO 15930 ELSE GOTO 15940
15930 PRINT "END"
15940 IF=0 THEN GOTO 15950 ELSE GOTO 15960
15950 PRINT "END"
15960 IF=0 THEN GOTO 15970 ELSE GOTO 15980
15970 PRINT "END"
15980 IF=0 THEN GOTO 15990 ELSE GOTO 16000
15990 PRINT "END"
16000 IF=0 THEN GOTO 16010 ELSE GOTO 16020
16010 PRINT "END"
16020 IF=0 THEN GOTO 16030 ELSE GOTO 16040
16030 PRINT "END"
16040 IF=0 THEN GOTO 16050 ELSE GOTO 16060
16050 PRINT "END"
16060 IF=0 THEN GOTO 16070 ELSE GOTO 16080
16070 PRINT "END"
16080 IF=0 THEN GOTO 16090 ELSE GOTO 16100
16090 PRINT "END"
16100 IF=0 THEN GOTO 16110 ELSE GOTO 16120
16110 PRINT "END"
16120 IF=0 THEN GOTO 16130 ELSE GOTO 16140
16130 PRINT "END"
16140 IF=0 THEN GOTO 16150 ELSE GOTO 16160
16150 PRINT "END"
16160 IF=0 THEN GOTO 16170 ELSE GOTO 16180
16170 PRINT "END"
16180 IF=0 THEN GOTO 16190 ELSE GOTO 16200
16190 PRINT "END"
16200 IF=0 THEN GOTO 16210 ELSE GOTO 16220
16210 PRINT "END"
16220 IF=0 THEN GOTO 16230 ELSE GOTO 16240
16230 PRINT "END"
16240 IF=0 THEN GOTO 16250 ELSE GOTO 16260
16250 PRINT "END"
16260 IF=0 THEN GOTO 16270 ELSE GOTO 16280
16270 PRINT "END"
16280 IF=0 THEN GOTO 16290 ELSE GOTO 16300
16290 PRINT "END"
16300 IF=0 THEN GOTO 16310 ELSE GOTO 16320
16310 PRINT "END"
16320 IF=0 THEN GOTO 16330 ELSE GOTO 16340
16330 PRINT "END"
16340 IF=0 THEN GOTO 16350 ELSE GOTO 16360
16350 PRINT "END"
16360 IF=0 THEN GOTO 16370 ELSE GOTO 16380
16370 PRINT "END"
16380 IF=0 THEN GOTO 16390 ELSE GOTO 16400
16390 PRINT "END"
16400 IF=0 THEN GOTO 16410 ELSE GOTO 16420
16410 PRINT "END"
16420 IF=0 THEN GOTO 16430 ELSE GOTO 16440
16430 PRINT "END"
16440 IF=0 THEN GOTO 16450 ELSE GOTO 16460
16450 PRINT "END"
16460 IF=0 THEN GOTO 16470 ELSE GOTO 16480
16470 PRINT "END"
16480 IF=0 THEN GOTO 16490 ELSE GOTO 16500
16490 PRINT "END"
16500 IF=0 THEN GOTO 16510 ELSE GOTO 16520
16510 PRINT "END"
16520 IF=0 THEN GOTO 16530 ELSE GOTO 16540
16530 PRINT "END"
16540 IF=0 THEN GOTO 16550 ELSE GOTO 16560
16550 PRINT "END"
16560 IF=0 THEN GOTO 16570 ELSE GOTO 16580
16570 PRINT "END"
16580 IF=0 THEN GOTO 16590 ELSE GOTO 16600
16590 PRINT "END"
16600 IF=0 THEN GOTO 16610 ELSE GOTO 16620
16610 PRINT "END"
16620 IF=0 THEN GOTO 16630 ELSE GOTO 16640
16630 PRINT "END"
16640 IF=0 THEN GOTO 16650 ELSE GOTO 16660
16650 PRINT "END"
16660 IF=0 THEN GOTO 16670 ELSE GOTO 16680
16670 PRINT "END"
16680 IF=0 THEN GOTO 16690 ELSE GOTO 16700
16690 PRINT "END"
16700 IF=0 THEN GOTO 16710 ELSE GOTO 16720
16710 PRINT "END"
16720 IF=0 THEN GOTO 16730 ELSE GOTO 16740
16730 PRINT "END"
16740 IF=0 THEN GOTO 16750 ELSE GOTO 16760
16750 PRINT "END"
16760 IF=0 THEN GOTO 16770 ELSE GOTO 16780
16770 PRINT "END"
16780 IF=0 THEN GOTO 16790 ELSE GOTO 16800
16790 PRINT "END"
16800 IF=0 THEN GOTO 16810 ELSE GOTO 16820
16810 PRINT "END"
16820 IF=0 THEN GOTO 16830 ELSE GOTO 16840
16830 PRINT "END"
16840 IF=0 THEN GOTO 16850 ELSE GOTO 16860
16850 PRINT "END"
16860 IF=0 THEN GOTO 16870 ELSE GOTO 16880
16870 PRINT "END"
16880 IF=0 THEN GOTO 16890 ELSE GOTO 16900
16890 PRINT "END"
16900 IF=0 THEN GOTO 16910 ELSE GOTO 16920
16910 PRINT "END"
16920 IF=
```



```

0010 SCORE=180
0020 A=1
0030 IF LEFT$(BL,1)="" THEN A=1
0040 LA-RIGHT$(LA,LEN(LA)-1):GOS
TO 180
0050 DO
0060 SCORE=200:COMMON:END:END:
GOTO 1
0070 SCORE=180
0080 LOOP UNTIL LEN(LA)=0
0090 LOOP UNTIL A=1 OR L=0
0100 IF A=1 GOTO 180
0110 PRINT "NO GAMING = UNSPECI
TED END OF RELATIONS, INCOMPLETE
"#:GOTO 2170
0120 FOR INCLUSIVE SECTION IS OPEN
END = NO ERROR IF PASSING
0130 DO
0140 INPUT LA,LC,LC2
0150 LOOP UNTIL LEFT$(LA,1)=""
OR LC=0
0160 IF LC=0 THEN 2170
0170 IF LEFT$(LA,1)="" THEN
GOTO 2050
0180 PRINT "NO GAMING ERROR = IN
FLUX EXPEDITED IN LINE":GOTO 1
10
0190 PRINT "NO REMAINING LINES I
N BOARD. = NO GOS 2170
0200 REM PROCEEDS APPLICATION LIT
ER = YES 3 / TRUE OR FALSE POSSE
S
0210 I=1:GOTO 2170:GOTO 1:IF FOR I=1
TO CN FOR J=0 TO L:GOTO 2:GOTO
NEXT:PRINT J=0:PRINT "ENDING INP
UTS SECTION AT LINE":GOTO 1
GOTO 10
0220 INPUT LA,LC,LC2
0230 IF LEFT$(LA,1)="" THEN G
OTO 2170:GOTO 1:IF FOR I=1 TO CN
FOR J=0 TO L:GOTO 2:GOTO
NEXT:PRINT J=0:PRINT "ENDING INP
UTS SECTION AT LINE":GOTO 1
GOTO 10
0240 SCORE=200
0250 DO UNTIL LEN(LA)=0
0260 SCORE=180:SCORE=200:IF LA=
LA+1 GOTO 2170
0270 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0280 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0290 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0300 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0310 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0320 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0330 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0340 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0350 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0360 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0370 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0380 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0390 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0400 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0410 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0420 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0430 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0440 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0450 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0460 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0470 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0480 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0490 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0500 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0510 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0520 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0530 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0540 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0550 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0560 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0570 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0580 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0590 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0600 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0610 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0620 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0630 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0640 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0650 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0660 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0670 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0680 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0690 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0700 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0710 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0720 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0730 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0740 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0750 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0760 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0770 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0780 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0790 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0800 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0810 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0820 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0830 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0840 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0850 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0860 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0870 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0880 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0890 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0900 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0910 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0920 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0930 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0940 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0950 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0960 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0970 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
0980 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170
0990 SCORE=200:SCORE=180:IF LA=
LA-1 GOTO 2170
1000 SCORE=200:SCORE=180:IF LA=
LA+1 GOTO 2170

```



DON'T LET THE AUSSIE BLUES GET TO YOU

POOLBLUSTER4 IS THE ONLY POOLS PROGRAM AT HOME "DOWN UNDER", AS WELL AS IN EUROPE. Every week-end it guides you through the rough terrain of the Aussie leagues to successful predictions over and over, again and again. It's a very successful conversion from the STAGE 27 program. Go on, don't hesitate, join the STAGE winners. It works for any game, cup or league, either British or Australian or European. It has a massive database, it's user friendly, it gives you a choice at the final completion of the coupon. Perfect for pools, ideal for fixed odds. Are you superstitious?

Do you believe in the power of the stars? If you answer 'yes' to any of the above 8 questions then POOLBLUSTER4 is for you. Its MACRO PREDICTION function will happily accommodate you. Do you believe in the power of the numbers? If the answer is yes, then POOLBLUSTER4 has something for you as well. Its "PREDICTION BY ELIMINATION" function will fulfil your exact requirements. And for the rest of us mortals, it will manipulate all the required information and it will give you different forecasts: a normal results forecast, a best house, a best drawn and a best away forecast. Are you into probabilities, percentages and bar graph representations? POOLBLUSTER4 will give them to you. In fact it has everything that you need for a successful forecast. It is truly the ultimate and the state-of-the-art pools program. Its hairy outer your coop today and win the pools tomorrow.

POOLBLUSTER4 COSTS ONLY £40!

SPECIAL OFFER: THE FIRST 34 ORDERS WILL RECEIVE A BOY GAME FREE. SO HURRY, DON'T MISS YOUR CHANCE TO BECOME A POOLS WINNER. WE ACCEPT ACCESS AND VISA CARDS.

MAKE CHEQUES / P.O. PAYABLE TO "APOLONIA SOFTWARE" AND SEND THEM TO: APOLONIA SOFTWARE, UNIT 12, PARK HOUSE, 148 BATTERSEA PARK ROAD, LONDON, SW16 4NN. PLEASE STATE IF YOU WANT CASH OR TAPE.

PHONE: 01 / 726-4400. FAX: 06 / 882-1861. ALL OFFERS ARE SUBJECT TO STOCK AVAILABILITY.

APOLONIA SOFTWARE THE COMPANY THAT CARES

PrintFx Correction

The Gremlins that caused havoc with the listing have been thwarted at last.

By Paul Eves



In the May edition of Your Commodore we published a Print Special Effects program for the Plus4.

Unfortunately, those dreaded gremlins reared their ugly heads and made a mess of the listings. You will no doubt have noticed some rather strange keywords and symbols. This was caused by the ROM in the printer

having a day off.

Hence then, for your pleasure, are the corrected listings. You will notice that listing 3 does not have the data statements on the end. (There is no point in re-typing these as they were correct originally.)

We apologise for this error and hope that it will not spoil your enjoyment of this excellent program.

```
10 REM *****
20 REM *
30 REM * PRINTFX -A BASIC LOADER
40 REM *
50 REM *****
60 REM
70 GRAPHIC 1,GRAPHIC 0
80 PRINT CORRECT"PRINTFX -A BN
90 LOADER PROGRAM"
90 PRINT"WRITTEN DECEMBER '80 BY
100 P.M. EVERSOMAN"
100 PRINT"*****
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990
```

```
1000 PRINT"*****
1010 LOADER *****
1020 NAME"PRINTFX LOADER",PROMO
1030
1040 PRINT"*****
1050 PROGRAM *****
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
```

```
10000
10001
10002
10003
10004
10005
10006
10007
10008
10009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
10100
```

```
10000
10001
10002
10003
10004
10005
10006
10007
10008
10009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
10100
```

```
10000
10001
10002
10003
10004
10005
10006
10007
10008
10009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
10100
```


Dos 6.1

We put everything you have learned in this series to some practice use

By Fergal Moane

This final article of the series presents a complete disk operating system, which puts into practice some of the theory already learned. If you are interested, disassembly of the code from SCDS-B onwards should be useful.

Dos 6.1 - The Utility

This utility is in the same mould as DOS 5.1 on the demo disk. DOS 5.1 is the most used Commodore utility in America. This system outstrips it in a number of ways:

- 1) DOS 6.1 actually techniques it's new commands. Tokens are one-byte values representing commands, and are used by ordinary Basic for all commands. This ensures compatibility, space economy, and no need for a prefixing character.
- 2) The above feature means that DOS 6.1 commands can be used in program mode, unlike most normal extensions.
- 3) DOS 6.1 provides more comprehensive commands in an easily understood form.
- 4) You are left around 3K free in the 4K block of memory from \$C000-\$D000.
- 5) Loading is under ten seconds, providing you save out the code separately, and the program is easily copied to your own disks.

Users Guide

DISK ERROR STATUS

DERR displays the status of the disk drive at the current position on the screen. See your manual for a full explanation of error messages.
Syntax: **DERR**

DOS8 COMMAND

DOS8 sends a command to the disk drive, opening up a range of around 50 disk commands. Again, see a manual for details of standard disk commands. If DOS is used without a

command, the error status will be returned.

Syntax: **DOS"command"**

DIRECTORY

Displays the directory of drive 0 or 1 (Usually drive 0) without loading it into memory.

Syntax: **DIR, drive number**

MEMORY SAVE

Saves memory from start address to end address to the specified device. This is very useful with graphics or machine code data.

Syntax: **MSAVE, start, end, "filename", dev**

DEFAULT TO DISK

This sets device 0 as the default device for all **SAVE** and **LOAD** operations. **LOAD"filename"** will now have the same effect as **LOAD"filename",0**

Syntax: **DISK**

DEFAULT TO TAPE

Tape is now the default device for **SAVING** and **LOADING**.

Syntax: **TAPE**

NORMAL DEVICE NUMBERS

This returns to the normal Commodore device numbering system. Use this if you need to load with **AJ**.
Syntax: **NORMAL**

LIST BASIC PROGRAM

DLIST allows you to list a Basic program directly off disk without loading it into memory. This is useful in checking a program, or grabbing lines from another program. **RUN/STOP** will stop the listing at any time.

If you try to list a non-basic program, you will get the expected garbage. Press **RUN/STOP RESTORE** followed by **CLOSE** to recover.

Syntax: **DLIST,"filename"**

LEAVE DOS 6.1

Leaves the DOS 6.1 system without disturbing Basic or any memory.

SYSTEM: SYSTEM61 will return you to the DOS 6.1 system.

Syntax: **QUIT**

MEMORY USAGE

A memory map of the system could be useful:

C000 - C120	Program for adding new columns
C130 - C1B0	Table of jump addresses for new commands
C200 - C53A	New Dos routines
D5B0 - CFFF	Free RAM for user programs

Some low memory and tape page locations are also altered during the command processing, but this has no effect on Basic and the area around the tape buffer is untouched.

Loading DOS 6.1

Type in and save the Basic program. When run, you will be presented with a screen of instructions. The machine code is being **FORed** into memory at this stage. On completion of the program **NEWS** itself and you are into the DOS system. Obviously, you will not want to keep having to load the basic loader each time you wish to use it. Therefore, I suggest you use the **MSAVE** command to save out the relevant two portions of memory.

USING DOS 6.1 IN YOUR OWN PROGRAM

As the commands can be used in your own programs, it would be inconvenient to have to load DOS 6.1 in every time, especially if your program is used by others. Follow the following procedure to include it in your program.

Load the two machine code files that you have saved at the start of your program. Immediately afterwards **SYSTEM: SYSTEM61** to initialize the commands.

Program Notes

When entering programs in the modified environment, **CH** Basic insists that you now have to enter colons after a **THEN** statement. There seems to be no way to override this, but it does not effect programs saved in the normal way. These will work normally.

EG:- **10 GET A\$: IF A\$="" THEN GOTO 10**

Also note that some commands will return no direct results if used in

a program. A bit of POKEing of the keyboard buffer could solve this. Unfortunately, this is unsatisfactory.

I hope that this series of articles has been of use to you, and makes your study of the subject less burdensome.

[illegible]

```

30 10 SET @MAYBE TO HAVE
31 20 SET @TIME TO ONE PAGE
32 30 SET @PAGE TO ONE PAGE
33 40 SET @PAGE TO ONE PAGE
34 50 SET @PAGE TO ONE PAGE
35 60 SET @PAGE TO ONE PAGE
36 70 SET @PAGE TO ONE PAGE
37 80 SET @PAGE TO ONE PAGE
38 90 SET @PAGE TO ONE PAGE
39 100 SET @PAGE TO ONE PAGE
40 110 SET @PAGE TO ONE PAGE
41 120 SET @PAGE TO ONE PAGE
42 130 SET @PAGE TO ONE PAGE
43 140 SET @PAGE TO ONE PAGE
44 150 SET @PAGE TO ONE PAGE
45 160 SET @PAGE TO ONE PAGE
46 170 SET @PAGE TO ONE PAGE
47 180 SET @PAGE TO ONE PAGE
48 190 SET @PAGE TO ONE PAGE
49 200 SET @PAGE TO ONE PAGE
50 210 SET @PAGE TO ONE PAGE
51 220 SET @PAGE TO ONE PAGE
52 230 SET @PAGE TO ONE PAGE
53 240 SET @PAGE TO ONE PAGE
54 250 SET @PAGE TO ONE PAGE
55 260 SET @PAGE TO ONE PAGE
56 270 SET @PAGE TO ONE PAGE
57 280 SET @PAGE TO ONE PAGE
58 290 SET @PAGE TO ONE PAGE
59 300 SET @PAGE TO ONE PAGE
60 310 SET @PAGE TO ONE PAGE
61 320 SET @PAGE TO ONE PAGE
62 330 SET @PAGE TO ONE PAGE
63 340 SET @PAGE TO ONE PAGE
64 350 SET @PAGE TO ONE PAGE
65 360 SET @PAGE TO ONE PAGE
66 370 SET @PAGE TO ONE PAGE
67 380 SET @PAGE TO ONE PAGE
68 390 SET @PAGE TO ONE PAGE
69 400 SET @PAGE TO ONE PAGE
70 410 SET @PAGE TO ONE PAGE
71 420 SET @PAGE TO ONE PAGE
72 430 SET @PAGE TO ONE PAGE
73 440 SET @PAGE TO ONE PAGE
74 450 SET @PAGE TO ONE PAGE
75 460 SET @PAGE TO ONE PAGE
76 470 SET @PAGE TO ONE PAGE
77 480 SET @PAGE TO ONE PAGE
78 490 SET @PAGE TO ONE PAGE
79 500 SET @PAGE TO ONE PAGE
80 510 SET @PAGE TO ONE PAGE
81 520 SET @PAGE TO ONE PAGE
82 530 SET @PAGE TO ONE PAGE
83 540 SET @PAGE TO ONE PAGE
84 550 SET @PAGE TO ONE PAGE
85 560 SET @PAGE TO ONE PAGE
86 570 SET @PAGE TO ONE PAGE
87 580 SET @PAGE TO ONE PAGE
88 590 SET @PAGE TO ONE PAGE
89 600 SET @PAGE TO ONE PAGE
90 610 SET @PAGE TO ONE PAGE
91 620 SET @PAGE TO ONE PAGE
92 630 SET @PAGE TO ONE PAGE
93 640 SET @PAGE TO ONE PAGE
94 650 SET @PAGE TO ONE PAGE
95 660 SET @PAGE TO ONE PAGE
96 670 SET @PAGE TO ONE PAGE
97 680 SET @PAGE TO ONE PAGE
98 690 SET @PAGE TO ONE PAGE
99 700 SET @PAGE TO ONE PAGE
100 710 SET @PAGE TO ONE PAGE
101 720 SET @PAGE TO ONE PAGE
102 730 SET @PAGE TO ONE PAGE
103 740 SET @PAGE TO ONE PAGE
104 750 SET @PAGE TO ONE PAGE
105 760 SET @PAGE TO ONE PAGE
106 770 SET @PAGE TO ONE PAGE
107 780 SET @PAGE TO ONE PAGE
108 790 SET @PAGE TO ONE PAGE
109 800 SET @PAGE TO ONE PAGE
110 810 SET @PAGE TO ONE PAGE
111 820 SET @PAGE TO ONE PAGE
112 830 SET @PAGE TO ONE PAGE
113 840 SET @PAGE TO ONE PAGE
114 850 SET @PAGE TO ONE PAGE
115 860 SET @PAGE TO ONE PAGE
116 870 SET @PAGE TO ONE PAGE
117 880 SET @PAGE TO ONE PAGE
118 890 SET @PAGE TO ONE PAGE
119 900 SET @PAGE TO ONE PAGE
120 910 SET @PAGE TO ONE PAGE
121 920 SET @PAGE TO ONE PAGE
122 930 SET @PAGE TO ONE PAGE
123 940 SET @PAGE TO ONE PAGE
124 950 SET @PAGE TO ONE PAGE
125 960 SET @PAGE TO ONE PAGE
126 970 SET @PAGE TO ONE PAGE
127 980 SET @PAGE TO ONE PAGE
128 990 SET @PAGE TO ONE PAGE
129 1000 SET @PAGE TO ONE PAGE

```

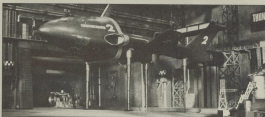
[illegible][illegible]

C O M P E T I T I O N

ENTER



No Strings Attached



Win wonderful prizes, care of Grandlam, in our easy-to-enter Thunderbirds Competition.

Five first prize winners - drawn after the closing date - will receive a Thunderbirds video and the game for Amiga or C64. Pens and posters are available for use

run-up.

All you have to do is write the contest answers to the following questions and send them, on the back of a postcard or sealed envelope, to:

Thunderbirds Compo, Your Commodore, Argus House, Boundary Way, Hemel Hempstead, HP3 7ST.

1. What is Brain's real name?

a) Horatio Hackenbacker

b) Bartholomew Baker

c) Cuthbert Cooke

2. What does FAB stand for?

a) Fantastic Automatic Breakthrough

b) Nothing

c) Funderbirds Are Brill

3. What is Parker's nickname?

a) Nasty

b) Shifty

c) Buzzer

Closing date: 31st August 1989.

The Rules

Entries will not be accepted from employees of Argus Specialist Publications, Grandlam, or Topex. This

restriction also applies to employees' families and agents of the companies.

The entry restrictions form part of the rules and the Editor's decision

is final. No correspondence will be entered into. In the event of a postal strike, we reserve the right to extend the closing date.

This Green and Pleasant Land!



Emma Norman dons her green outfit and, with banner grasped firmly in hand, campaigns for the organisation that everybody should know about

In a recent advertising campaign the organisation Friends Of the Earth gave five billion reasons why you should join them: The Human Race. These same five billion reasons could be given for joining the environmental organisation Greenpeace. Both Greenpeace and Friends of the Earth have the same object at heart:

To conserve the planet for future generations.

Most people have heard of Greenpeace and are aware of the work they do, but for those who do not know it is, in its own words: "An international Environmental Pressure group which maintains complete independence from all political parties anywhere



in the world. Its aim is to bring an end to the pollution of the planet and to protect wildlife." It operates throughout the world.

It began in 1971 in Canada, with a protest voyage into a nuclear test zone at Amukika. The test was disrupted and the area is now a bird sanctuary. In 1973 Greenpeace opened its first branch in Britain where, to date, it has over 120,000 members. Today there are over 340 million members Worldwide with the majority of these coming from the USA. This figure is one of which they are very proud, but many members are always needed. Despite the successes, there are

many objectives still to be reached.

Why do we need Organisations like Greenpeace? What is so harmful with dumping waste? The wastes are concentrations of poisonous substances: metals like Copper, Lead, Zinc, Chlorine, Mercury and Arsenic. These build up in the environment and poison food sources. For example: in Wood Spring, North Devon people were worried not to eat more than 1lb of Shrimps or 4oz of shellfish due to the excess of cadmium in harmful poison/theta. Two million tonnes are year of such waste is dumped in the Irish sea.

Although a direct connection has

not been established between waste-dumping and the virus that killed 17,000 seals last year (and appears to be re-emerging amongst the remaining Common Seal population), the possibility that pollution was the cause cannot be overlooked and emphasises the urgency for action. There are many more examples of such atrocities caused by "Man's inhumanity to man", including Chernobyl, the Alaska oil spillage, the plight of the whales, elephants, too numerous to list, but what of the successes?

In the past five years Greenpeace has stopped hunts of seal pups; helped bring about an end to commercial whaling; stopped the burning of hazardous waste in the North Sea, US waters and the Mediterranean; stopped the dumping of radioactive wastes at sea; helped persuade the British Government to spend millions of pounds cleaning the beaches in Britain and filtering the waste from coal-fired power stations.

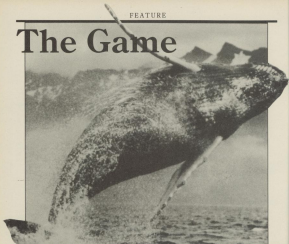
The majority of these achievements are brought about by sheer bravery. Greenpeace activists went out in dinghies and positioned themselves between the harpoons of the whaling boats and the fleeing whales. As a result Commercial whaling is now banned. In the North Atlantic, Greenpeace drove its inflatable dinghies under barrels of radioactive waste, therefore dumping of such waste at sea has now been stopped. Volcanoes physically prevented baby seals being killed in the Orkney Isles and New-foundland by throwing themselves between the hunters and the pups.

The level of the danger in which these people place themselves became apparent when one man was killed on the Rainbow Warrior when it was destroyed whilst trying to stop the testing of nuclear weapons off the coast of New Zealand. These people are risking their lives? What can you do to help?

You can join them in their fight to protect the planet by becoming a member, information and membership forms are available, from Greenpeace 20-31 Blenheim Green, London N1 8BN. (Subscriptions cost £12 for a single, £17.50 for the family, £6 unwaged, and £20 for overseas).

Greenpeace is at present campaigning for Nuclear Free seas in Greenpeace, Nuclear Free Irish sea, stopping toxic waste dumping in Spain, working to protect the seals and dinghies round the British coast, and trying to get Australia declared a Waste Free.

The Game



ENTITLED 'Rainbow Warrior', the Greenpeace game is set to raise a few eyebrows. It has already provoked a Conservative MP to condemn it, believing that many computer gamers may act about industrial pipelines and use other forms of violence in order to 'practice' the Greenpeace methods shown in the game.

The gameplay is split into several sections, all of which are based on separate ecological problems. You play a Greenpeace supporter who is determined to protect wildlife and stop pollution of the natural world, and each section is a mini-game all in itself. Among other tasks you have to prevent whaling ships from catching their prey by manœuvring the Greenpeace boat

between the whale and the ship, save penguins from straying into a beam that has broken through the ozone layer, and you must stop sea-culling by ruining the pelt of seals; this is done by spraying them with green paint.

Although the game features very strong opinions, Macropose has made the graphics slightly humorous in tone, so as to appeal to a wider market.

The sprites are enormous and the animation as smooth as baby's bottom; they have even been described as Pollockesque.

Rainbow Warrior should soon be available on Amiga and C64, and it is to coincide with the new Greenpeace album, Rainbow Warriors, which features such artists as U2 and Simple Minds.



RAINBOW WARRIOR

RAINBOW WARRIORS

The world's first environment friendly software!

At last, an all action game that presents a solution to the environmental dilemma faced by the entire human race. Rainbow Warriors is an action game with real depth and meaning. It simulates seven campaigns of Greenpeace members over the last few years.



*Micro
Style*

GAMES FOR ADULTS

THERE have been many cult shows from television. By 'cult' I mean shows which have seen more repeats than a particularly bad curry. Examples are *The Wooden Tops*, *Bill and Ben - The Flowerpot men*, *Andy Pandy* and *Thunderbirds*. All of these have something rather significant in common, their actors were more wooden than the cast of *Neighbours*, and have more strings than all of *Witchhick's* Eerie Children put together. They all featured puppets.

The latter of these repeatable shows is the one which we are to spotlight here, mainly due to the fact that *International Rescue* use their first pixellised exploits, thanks to software company *Granadeira*, whose previous credits include *Parsons*, *Flowermen* and *The Running Man*, as well as the forthcoming *Livelihood - The Computer Football Game* (which has been widely acclaimed). *Thunderbirds* is its first venture into the world of Gerry Anderson, although contacts have it that it will not be the last (look out *Nymrocks*...!).

Released in the early sixties, *Thunderbirds* gained popularity at an alarming rate, due mainly to Anderson's previous successes - *Captain Scarlet*, *Fireball XL5*, and *Stingray* to name a few - but, there was something rather special about the Tracy family that enthralled viewers everywhere, and the plots were far more detailed than the simple rescue missions that they seemed to portray.

The characters created were not as shallow as one may expect. Each one had a more complex background and history than many at the forefront of soap operas, thus making for a slightly more involving program. The personnel of *International Rescue*



THUNDERBIRDS

runs as follows:

Jeff Tracy - Founder of *International Rescue*, and father of five sons (who are also in the 'family business').

Gerry Tracy - Eldest of the brothers, pilot of *Thunderbird 1*, and second in command if anything happens to his father.

Pilot Tracy - Pilot of *Thunderbird 2*, also deputises on *Thunderbird 3* when Alan is not available.

Alan Tracy - Pilot of *Thunderbird 3*, also alternates with John on tours of duty on *Thunderbird 5*. Is romantically involved with *TiaTia*.

Gordon Tracy - Pilot of *Thunderbird 4*.

John Tracy - Spends most of his time on *Thunderbird 5*, due to his duty as a radio ham. Occasionally replaces Alan on *Thunderbird 3*. **Braine** - Designer of the *Thunderbird* equipment and technical genius. Has a tendency to stammer.

Kyrano - The Tracy island cook. Is half-brother of the villainous Hood.

TiaTia - Kyrano's daughter. Is in love with Alan, and helps Braine maintain and repair the *Thunderbird* equipment.

Lady Penelope - Ex-British agent, famous for her unique pink *Rolls Royce* PAB 1.

Parke - Failed sales cracker who was caught by Lady Penelope. Acts as her butler as 'cover'.

All of these except Lady Penelope and Parke were based on Tracy Island, which seemed audiences with its great technical sequences when the *Thunderbirds* lifted into the air, more memorable of which was the swimming pool which did back to allow *Thunderbird 1* to ear from its underground launching pad. The craft themselves were superbly designed, thus encouraging kiddies everywhere to request a home version of *Thunderbird 2* - the most ingenious of them all - courtesy of *Etiky Toys*. *Thunderbird 2* was popular because of it's ability to store other pieces of equipment - namely *Thunderbird 4* - in its stomach.

It is this depth, characterisation, gripping plots and technical brilliance that made *Thunderbirds* the most successful of Gerry Anderson's productions, and it is for this reason that the computer game should not go without notice. The game has many of its own merits, being the best arcade adventure to see



these shows for many months in just the first.

The game features four different missions. You must control two characters in each and perform certain actions in order to come out unscathed (although Thunderbirds can never die, and so again the lead Anderson) and acclaimed as a hero. The first - Mine Mine - involves Alan and Brian who have to rescue some trapped miners. This is not a simple operation as priority lies in shutting off the mine's pumping system before it floods the whole mine, thus rendering all to be rescued by a rival team, causing embarrassment and chaos, to work either help - like I said, Thunderbirds cannot die.

The other three levels - Sub Crash, Bank Job and Countdown to Terror - respectively - are all in a similar vein, although time limits in which to complete your task get progressively more restrictive, and the problems connected to the mission become harder to figure out.

Graphics throughout are very pretty and true to Anderson's original ideas. Presentation throughout is of a very high quality, for instance, before each mission there is a small digitised animation sequence from the old episodes, for the four missions are all based on existing, single shows, although only loosely in those who own the videos.



will find no favours in shouting. The famous theme tune - 5... 4... 3... 2... 1... Thunderbirds are Go!... Boom... Boom... Boom... etc. - has been reworked to a rather good house version, which accompanies the title sequence but leaves the game itself to sampled sound effects.

Another small thing that I think helps sway my attitude about this game is the small animated sequences which the graphics go through at certain stages (similar to Little Computer People) giving the game more character. Each mission has a password which once discovered allows passage straight into the section without the earlier wussy



work I think. Payne, Grandblam, and Gerry Anderson will all be very pleased with the final product of some draft looking and hard work. I know I am as a lifelong Thunderbirds fan and I can see myself getting as gripped by this computer version as I was with the original series. Although it really is one of the oldest shows in the book this time, for me, Thunderbirds ARE gold!

Bill Henderson

Payne are the programming team behind Chubby Goats, Tossamen, and the more recent Prison - proving that they are really arcade adventure experts - although Thunderbirds is the most complex of all, and probably the most complex.

Grandblam - £24.95



INFO

Gameplay: 95%
Graphics: 95%
Sound: 95%
Usability: 85%
Overall: 92%



GILBERT - ESCAPE FROM DRILL

Again Again - £9.99/£14.99

Follow Gilbert on his madcap escapades as he explores the planet Drill, whilst searching for the missing parts to the Millennium Dacthion.

THE cute but snooty alien, Gilbert, is hankering once again for the bright lights of ITV. He has the contract, and the ego, now all he needs to do is find the missing parts of the Millennium Dacthion and sign on the dotted line.

Gilbert has twenty-four hours to find the five missing parts, hidden by the discerning TV watchers of the planet, before some other, hopefully more talented, star signs the contract. No Drillies possesses a full set of faculties and those would-be submitters have actually left clues in various video games, which should help Gilbert fulfil his quest in mortifying the minds of the

Earthly young.

Gilbert starts his adventure in the middle of his home town, though why someone would wish to build a town in such a hostile place is beyond me. As you drift off - yes Gilbert has no feet - across town you are assaulted by all manner of beasts.

Fortunately Gilbert can dispatch these creatures with a flick of the head and a globe of acid. This formidable method of defense is finite but fresh supplies of the sticky green stuff can be found in milk bars, along with the video games which gave the missing parts. If Gilbert manages to shoot an entire wave of creatures, a flying

plane of wobbling jelly grenades its way onto the screen. Shooting this either liberates nothing, a can of beans, or a piece of cake. Of which four of these items can be carried at any one time. If the beans are eaten Gilbert bleeds and floats, enabling him to reach the upper levels of the forest and change levels when in the sewer system. Eating the cake brings Gilbert back to Drill with a bump, obviously made with the wrong brand of flour.

The five video games which Gilbert must complete offer a welcome

break from the more

mundane pleasures of spotting monsters in the street. Some of the better games include Earth Invasion - in which a one-eyed monster blasts at rows of descending humans. Brain Drain - play against the computer to see who can match the most tiles. And Spotlight at the O4, Corral - spot 'em cowboy. Failure to complete a video game produces a loss of one hour of time.

Unlike other games the attacking creatures never cease to appear as they come forth in a never ending variety of pulses. The playing area is not that small, and not only is there a microscope to control with, but a jungle, cave, tree top and underwater adventures to occupy the mind. Seemingly good fun even if the gameplay is a little too repetitive.

Adrian Pomphrey

INFO

Gameplay: 65%
Graphics: 82%
Sound: 78%
Usability: 88%
Overall: 79%

Enigma Variations, headed by Mark Garmakchik and Richard Taylor, are the people responsible for the latest TV presenter computer game and as a publicity stunt they sent all the larks a free sticky Gilbert to throw around - while fun that was.



SCRAMBLERS. In my opinion, could be so called because a couple of hours on the back of one of these things is a scrambled brain for me, dear. The only advantage posed here is one of less excitement coupled with even less pain than the real thing.

Fifteen tracks of pure frustration lie before you, and failure to complete one will be the last least means a trip all the way back to the beginning. You start with the three dirt tracks, beginning with the over so easy track 'A', which is designed to allow you to save time by letting you complete at least one of the fifteen tracks. To complete the first track you simply have to ensure that the gear you are currently in (out of three) is correct for the slope you are travelling on. And that the bike you have the purpose of controlling

does not exceed one of the two extremes of speed - stalling, or going so fast you end up flying over the handlebars.

On the second dirt track you hit your first real obstacles, and find a use for the other controllable movements; the raising of the front and back wheels - just as the same time I had to add. This track also teaches you the precise times which this game is hard, precise control over the bike's speed. When travelling over bumps and rubble you must ensure that your speed does not exceed a narrow set of limits. Failure to do so results in the action freezing and a message informing you whether you were too fast or too slow. You may note that you do not even have the satisfaction of seeing your bike hit the dirt.

By the time you have reached the third track you must have mastered the raising and lowering of the wheels - if not you might as well turn your computer off and start again. Here the main obstacle comes in the form of small holes, which require you to raise your bike wheels whilst passing over them. Next come three tracks with bottles, the car variety, strips and watery pits. On these courses the control over your speed becomes too critical for your joystick to handle with any degree of accuracy, and you inevitably crash.

If speed and gear selection were all you had to worry about, then life might then approach something like simple, but you also have to ensure that you stay on the course as indicated on a small strip of the screen which shows sideways movement.

If you are a masochist who enjoys the sound of revving engines then you might enjoy this game - but I doubt it.

Adrian Humphrey

Gremlin Graphics, once part of the Birmingham Software Syndicate, are known to many Sheffield where, hopefully, new ideas will reign.

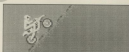
INFO

Category: 39%
Graphics: 83%
Sound: 88%
Longevity: 87%
Overall: 54%

SUPER SCRAMBLE SIMULATOR

Gremlin - £9.99/£14.99

Mount up and pit your sanity against the latest game to bring you all the thrills and spills of high risk sport without the risks and without the thrills and even less of the sport.





RED STORM RISING

Microprose - PC - £34.99

Tom Clancy's novel of World War III submarine warfare comes to life as you take the helm of a US Attack submarine. Sooner or later, it had to happen.

WHEN Islamic fundamentalists destroyed the Russian oilfields, war looked inevitable as the Great Bear set its sights on the Middle East. The combat that followed as the super powers clashed would be partially decided by naval actions in the crucial North Atlantic. As a submarine commander your strategy and skill will be vital in hunting and destroying enemy subs, task forces and troop carriers.

Before you take on such responsibilities you can test your skills in a series of training missions, during which time you can learn how to use the sonar thermal layer to avoid enemy detection, the importance of keeping your speed low and a reasonable depth to ensure your propeller doesn't cavitate in the water (and alert enemy subs), and how and when to use the weapon systems. The toughest part of submarine warfare is tracking down your enemy without him finding you. This means you can't use active sonar, which would give away your position. You must find a quiet part of the ocean and listen for the signs that your possible enemy can find. Once you have a good sonar contact (about 90% - many torpedoes are fired in bat-

tle poor contacts) you can fire your torpedoes at a submarine, or missiles at the surface contact.

Although you can play individual scenarios, the real challenge is when you play the Red Storm Rising Scenario and play your part in World War III. Here your missions are carried out and directly affect the backdrop of the war. Graphic sequences set the scene as NATO battles against the Soviet onslaught. An example of this is when you're sent in to intercept a task force carrying troops. Not only do you have to track down and attack the convoy, your success at sinking the troop carriers will decide the success or failure of the Soviet attack. If you do badly, the map will rapidly turn red.

As an idea, Red Storm Rising can't be faulted. It's based on a best selling book by a bestselling author and the computer implementation provides all the options and problems that face a modern submarine commander, who must do his best to win battles that affect the outcome of the war. Unfortunately, the game falls down on presentation. Although the clips showing the progress of the war are good, the actual screens

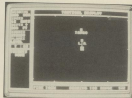
used in the battles (most of the game) are dull. In fact, there's no comparison to games such as Silent Service or 688 Attack sub. As a result this will deter many gamers from a challenging and fascinating game.

Tony Hetherington

Red Storm Rising was written, in conjunction with Tom Clancy, by Sid Meier who was the author of P-13, Justice Eagle and Silent Service. He was also the co-founder of Microprose.

INFO

Gameplay: 84%
Graphics: 32%
Sound: 18%
Usability: 67%
Overall: 53%



Islamic religious radicals outstage Russia's hopes of refinery expansion as Soviet interests in America. This destroys one third of the nation's production capacity. The Soviet Union faces years of crippling energy shortages and an emboldened military.

STORMLORD

Hewson - £14.99 Disc, £9.99 tape.

AN evil Queen dominates the once beautiful and peaceful land. It was a land where fairies frolicked and witch horses lay around being bitch. It was also incredibly dull, so it's just as well that the Queen arrived with her evil minions. Wicked creatures roam the land, man-eating plants have taken root, and frolicking fairies lie captured in cages. Now Stormlord must rescue the fairies and liberate the land by destroying the Queen.

The land is represented on screen by a series of levels, each consisting of a number of sideways scrolling screens.

Decile plants, statues, steps, and boulders form the platforms on which our hero can walk and jump. Control is somewhat

limited although you can build up power for a bigger leap by holding down the fire button. The screens also contain a very useful but inexplicable network of springboards that can haul our hero quite remarkable distances, and quite often land him by a dunnell in distress. You can also be sure that there will be another springboard strategically placed for the return journey.

While Stormlord is running, jumping and leaping about the land various nasties are out to get him. Huge worms want to nibble him. Dragons want to fry him, fire want to roast him and Venus fly traps want to chomp him. However, he can hit back with a self loading. Being roard, but it's an ineffective that it's best to get out of the way.

Fairy folk and foul creatures clash in a game that aims to take your screen by storm, as the heroic Stormlord rouses rebellion against a malevolent monarch.

There are also a number of traps such as boulders and acid which drop down from above and doors which block your path throughout the game.

Luckily, there are useful objects scattered around the landscape, but our hero and incredibly bitch hero can only carry one at a time, which I suppose isn't that surprising when you consider that a key is almost as big as the door it opens! Apart from the keys,

there are other objects whose function is less obvious, which can make it difficult to always carry the one you need next, resulting in quite a bit of backtracking. This can be reduced by leaving them near springboards.

The result of all this is a game that's a cross between a sideways scrolling shoot-em-up and a platform game. Although I don't think this will appeal to either game's supporters as it is

INFO

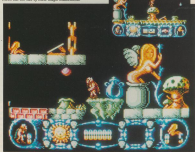
Graphics: 37%
Sound: 49%
Control: 82%
Longevity: 38%
Overall: 41%

rather silly and equally uninspiring.

Tony Hetherington

Stormlord was originally written by *Wagstaff Circus* (UK conversion by *Nick Jones*), who also produced *Cyberoid*, *Cyberoid II* and *Exolon*.

Freedom man! Not only is this child frolicking naked, but check out the size of those magic mushrooms!



ROCKSTAR ATE MY HAMSTER

WITH all the sweat and toil of making it to the top, Rockstar Ate My Hamster provides a gritty insight into the topsy-turvy world of rock 'n' roll management. Through the eyes of a somewhat slimy hunk of no-talent

Codemasters - £9.95 Tape

Sex, drugs and Rock 'n' Roll - one of these is featured in Codemaster's latest full price game. Does the game live up to its rather peculiar name?

Gameplay: 90%
Graphics: 85%
Sound: 80%
Usability: 85%
Overall: 80%



straggler, called Cecil Pitt, you act as a manager, decking and driving and calling the shots on the band's future plans. But getting to the top is not so simple.

The ultimate goal of Rockstar, which has the intriguing feeling of a board game, is to go Platinum by topping the charts, the world and elsewhere. With your lurching sidekick Clive, who insists on using groovy talk, your first task is to choose the band. You are allowed up to four members and can select from a list of about 50 stars who appear one by one on a video screen in front of you. It is probably best to choose the maximum of four band members as somewhere along the road tantrums erupt in and the odd terrorist attack tends to deplete the ranks. As for selecting stars, which are loaded from a data cassette, it varies from player to player depending on who can handle the big money stars and tighter budgets.

Each star bears a distinct likeness to some of the big names in the pop world - including Duran Duran, Buggles (Kylie Minogue), Drex (Bruce) and many more. With your band together it's time to buy the gear. New equipment costs money but it's one step to true professionalism, with amplifiers that go to eleven for that extra push over the cliff. However, being a leader is not to everyone's

taste so there is the option of second-hand gear, or even lower to the dodgy knock-offs.

This is where your climb to the top or low-life stroll into the wilderness begins. To start with, you have four options: practice, gig, publicity or gifts. It is probably advisable to lock them away for a bit of a jam, give them some publicity and then hit the road. The practice sessions cost money, so you have to choose your studio time wisely. The publicity, which is dubiously handled by the scoundrel Clive, involves the aptly named Star Newspaper and includes a subtle reworking of that infamous headline concerning a hamster. Publicity stunts, however, do not always get covered by the press, but that's a chance you have to take.

If you decide to gig you have to select a venue from the list of pubs, clubs, Universities, Concert Halls and stadiums. For the first time out it is probably best to play to cheap and easy-to-fill venues. A decision on the ticket price has to be made and the number of nights playing. After each gig Clive informs you of the number of tickets that were sold and consequently the amount of money made. At the start of the game money is the key to success especially if you have chosen band members with high salaries, they have to

be paid on a weekly basis.

If gigs are going well and you have succeeded in front page guitar news, the phone will start to ring with offers of record contracts, charity gigs and potential sponsorships. When a satisfactory record deal is reached you have the option of ending the slay and shooting a promo to begin your claim on the record sales market and a position in the national charts. The prospect of shooting a video means more decisions on producers, locations and image details with every decision being of paramount importance for the band's future. With some suspiciously recognisable tunes the band under your guidance will enter the realm of the weekly charts and face the problem of staying at the top, paying taxes and using bookkeepers.

Rockstar Ate My Hamster is an enjoyable game even though the graphics are not particularly good. It is addictive with reasonably good sound for its calibre and price, and for this reason alone it is worth considering.

Mark Jones



Claire Jones, the programmer of all the formats of Rockstar, is undoubtedly, and without disguise in any way, Welsh.

ULTIMA TRILOGY

*Origin Systems (Microprose) - PC £29.95.
C64 £24.95 disk only.*

More than one million global gamers have delved into the Ultima series of role playing games. Now the first three are available in a special edition.

THE Ultima Trilogy is a special edition of the infamous Ultima games in which you must confront a Triad of Evil who threaten the land ruled by Lord British. Both Ultima I and III have been available in the UK before but this is the debut for Ultima II, which will provide many more hours of role playing, whilst filling some of the gaps and answering some questions that were left open with its cousin.

The content of the box is impressive and immediately gives the impression that this is a game to play. Alongside the game disks (3 for C64, 4 for PC) there is a quickstart guide, a player reference guide that outlines the keyboard controls and available spells, and a massive 180 page instruction manual that covers everything from the back of the author's neck to a guide to the monsters and enemies you will face. It also contains three full colour maps of the lands you must save.

In the games you play a brave adventurer who may be of human, elf, dwarf or halfling stock, and be trained in the art of a fighter, cleric, thief or wizard. Throughout the games your hero is controlled through a series of single key presses that correspond to commands. For example, 0 lets you offer gold as a bribe. U unlocks doors if you have the key, and I ignites a

tank. Although you begin alone with a number of points to distribute between your role playing characteristics, you will soon find others that will join you in your quest.

Ultima I is the first age of darkness and features a desperate battle against hordes of nightmarish creatures spawned by the Wizard Mordilain. The delinquent people of Ansania flee in vain to the ancient strongholds and all will be lost if a hero is not found.

The same heroic abilities will be put to the test in Ultima II when Minax, the young apprentice of Mordilain, seeks to avenge her father's death. Such is the power of this Enchantress that her efforts to find her father's slayer have torn rifts in the fabric of time. This has opened doors through which a brave adventurer might bring about her doom.

In Ultima III the third part of the Triad of evil is revealed, as fragments of a manuscript hint at an uneasy alliance between Mordilain and Minax and of a demonspawn that their union has produced. This is the ultimate test for a hero as evil openly stalks the land and your fate is unknown apart from the power is poisonous.

The games look a little dated now compared to the more recent Ultima IV and V, and other games such as *Fall of Britannia* and

INFO

Compatibility: 100%
Graphics: 45%
Sound: N/A
Usability: 80%
Overall: 72%

INFO

Compatibility: 75%
Graphics: 55%
Sound: N/A
Usability: 80%
Overall: 71%

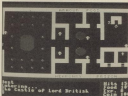
Red's Tale III, but this special edition is a must for Ultima players looking to complete the story.

Tony Hetherington
The Ultima adventures were written by Richard Garriott between 1979 and 1988, firstly being published

through Sierra-on-Line and then through Origin, the company he co-founded. The first Ultima game to appear in the UK was imported by US Gold (Ultima III) since then others have been imported by Microprose.



THE ULTIMA TRILOGY
PC 3.5" DISK 4 DISKS
C64 5.25" DISK 3 DISKS



Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session. Littering one of the superb programs from *Your Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

...Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Your Commodore Software Service* makes available all of the programs from each issue on both cassette and disk at a price of \$6.00 for disk and \$4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFOSNET LTD, 3 River Park Estate, Berkhamstead, Herts, HP4 1BL.
Tel: (04427) 15681

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C64 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strip containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strip. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the programs, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

APRIL 1989

BASIX - Give your C64 new sound, graphics and toolbar commands as well as a machine code assembler with this Basic extension.

AUTOSCROLL - Professional text screen scrolling with this C64 utility.

BALANCE SHEET - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).

ICBM 64 - Add icon control to your own programs with this C64 utility.

80 COLUMNS - A suite of Hi-res graphic commands for C128 80 column users.

ZEUS - A disk file copier for Plus/4 users. Available on disk only.

MULTI-PRECISION ARITHMETIC - A suite of powerful mathematical routines expanding your C64's number crunching abilities.

STORAGE SPACE - Use the RAM behind the C64's Kernel and Basic ROMs for data storage (available on disk only).

ORDER CODE

DISK YDAPR89 \$6.00

TAPE YCAPR89 \$4.00

MAY 1989

ANTI-FREEZE - Protect your C64 programs from cartridge-based freezes.

MULTI-COLOURED LIST - Brighten up your C64 Basic listings by adding on-screen colour.

PRINTFX - A suite of powerful screen printing commands to add to your Plus/4's Basic. (disk only).

BANKER 128 - Money Management for C128 disk users.
TURBO TOUCH - A superb typing trainer for C64 disk users.

ORDER CODE
DISK YCMAY89 £6.00
TAPE YCMAY89 £4.00

JUNE 1989

RETURNER - A handy routine that allows your C64 programs to re-load a disk based menu. (disk only)
LINE INPUT - Improve your C64's INPUT routine
PROGRAM ANALYSER - A suite of programs to help you examine your C64 Basic programs in depth.
ASM ASSEMBLER - A powerful 6510 assembler for the C64

ORDER CODE
DISK YDJUN89 £6.00
TAPE YDJUN89 £4.00

JULY 1989

MOUSE 80 - Use a 1251 mouse in 80 column mode on the C128.
QUICK SEARCH - Add a FIND option to C64 Basic.
SPRITE LIBRARY - Add a collection of sprites to your growing sprite library (C64)

HEAD FOR HOME - An update on our popular C16 and Plus/4 game. Available on disk only.
WINDOWS ON THE 64 - Powerful windowing commands for the C64.
SPRITZ SPRITE - Give your Plus/4 or expanded C16 sprites - just like a C64. Available on disk only.

ORDER CODE
DISK YDJUL89 £6.00
TAPE YDJUL89 £4.00

AUGUST 1989

REASONING ON THE 128 - Programs for the first part of our expert system series (128 Disk only).
THESAURUS - Build up your own C64 thesaurus with this program for disk or tape.
WINDOW ON A MAZE - Generating mazes on your C64
ADDRESS BOOK88 - A cassette based address book for C64 users
DISK SCRAMBLER - Protect your disk data from prying eyes (C64 disk).

ORDER CODE
DISK YDAUG89 £6.00
TAPE YDAUG89 £4.00

Cassettes or disks are available from March 1988. Please ring the editorial office (01-437 0626) for details of these.

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
SEPT 1989		TAPE (£4.00)	YCSEP89	
SEPT 1989		DISK (£6.00)	YDSEP89	
OVERSEAS POST £1				
			TOTAL	

NAME

ADDRESS

POSTCODE

I enclose a cheque/postal order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.

All orders should be sent to: YOUR COMMODORE

READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP1 7ST

Please allow 28 days for delivery.

TERRAMEX

Grand Slam - £2.99 Tape

High in the icy vastness of space, something is headed our way. A giant asteroid is on collision course with Earth and only you can save the world.

WELL, actually the only person who can save the world is Professor Eyrstrain, who predicted the asteroid's coming some 30 years ago. The trouble was that no-one believed him, so he went off to a dark and dingy continent to sulk.

Taking the part of one of five stereotyped explorers you must search out the lager of Lemel, and then search out Professor Eyrstrain so he can build an asteroid deflector. Although if the world is saved then Grand Slam will be able to write more programs like this... Hmm, total destruction might not be so bad an idea after all.

When Terramex first appeared on the ST and Amiga it had the feel of an average 8-bit game, and now we see it in budget 8-bit form, which is better value for money. It doesn't make the game any more interesting though.

Having selected your character and waited for the rest of the game to load, you set off into the dark continent followed by your invisible hangers, in undramatic 3-D arcade adventure style. An irritating piece of music will have your teeth grinding by now, so go for the volume switch whilst concentrating on the smailish but colourful characters, and the generally dull

landscapes.

One of the first objects encountered, and this game is object intensive, is a vacuum cleaner, which undoes the power of flight. Yes, you'll believe a Hoover can fly, but not everywhere. One of the failings of Terramex then becomes obvious, you can hover along a certain distance and suddenly because it would make the game too easy, you stay in mid air, unable to progress further. There isn't actually anything there stopping you from going on, you just can't. One of the next objects needed is an umbrella which acts as a parachute. One good point is that you can ask your character for an idea when faced with a problem. If you're lucky the right object will flash up, and then it's down to you to use it properly.

Besides the object manipulation part, there's also the challenge of having to leap over snakes, crevices, and other obstructions, in an lady Jones like manner. Should all efforts fail, then a quite humorous sequence follows whereby the asteroid plunges into the Earth. For sheer awfulness this takes some beating.

For aficionados of the arcade adventure, £2.99 represents good value, if unimpeccable entertainment, but for those not converted to the cause, there is nothing here to convince that an expedition to find Professor Eyrstrain is a better idea than waiting at a bar for disaster to fall.

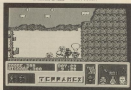
Duncan Evans

INFO

Concept: 60%
Graphics: 75%
Sound: 67%
Largibility: 71%
Overall: 68%

04

Grand Slam started life quite shakily, with Terramex being one of its poorer releases. Recent offerings are generally much better.



THE land has always been ruled by one of the four elemental magi. At the time of the great conclave, the Fire Wizard happened to be king, but news soon spread through the town that he has been horribly murdered. The Earth Mage has fled, the Water Mage has disappeared and the Wind Mage has been grievously wounded. Rumours of an enormous magical beast abound as food ripples start to run dangerously low. In desperation, the mayor sends forth a hero who promptly disappears. Now, you have been summoned to help.

You have the choice to play one of six characters, each differing in their strength, armour and magical abilities. These attributes can be altered temporarily with the help of magic or permanently by

finding relics - something that you will have to do if you are to get very far in the game.

Fire King follows the usual role playing ideas as so much as you have to wander round fighting monsters and finding key objects. There are books scattered around containing clues, although some information, especially in the Thieves Guild, has to be paid for.

The key to the game lies in manipulating the objects that you find and/or purchase. You have seven pockets each of which can contain up to nine similar objects, eg. keys, death spells, etc. The problem is that there are usually more than seven types of objects to carry, so you will need to plan ahead. On top of that, there is food to buy which will help to top up any energy lost in battle.



FIRE KING

Electronic Arts, Price £18.99 Dhr
Role-playing down under is the name of the game. Gordon Hamlett casts a clairvoyance spell and determines if everything is fair dinkum or not.



Graphically, the game looks and plays a lot like Gauntlet with the top-down view and monster-producing windows. On top of that, there is more of a feeling of adventure to the game rather than just one large shoot-em-up.

The problem with Fire King is the manipulation of objects. The pocket system is very cumbersome to use, especially in a 'real-time' situation. The monsters are everywhere and I suspect that fans of role playing games will find this too much of an arcade game for their liking. I must confess that knowing their previous track record, I expected something a bit more cerebral from this company.

Gordon Hamlett

INFO

Gameplay: 40%

Graphics: 51%

Sound: 41%

Usability: 30%

Overall: 50%

Written by an Australian Company called Micro Fante and Licensed through SSG - their first venture away from Warcraft.

JOURNEY TO THE CENTRE OF THE EARTH

US Gold - £19.99

Face danger, excitement and the unknown as you attempt to follow in the steps of 19th Century explorer Awe Sakmussen, that lead to the centre of the Earth.

If you want to explore a game based on Jules Verne's journey to the centre of the earth, programmed by the experts that wrote *Jour de l'été*, you certainly wouldn't imagine this game. It's not the fact that the game encompasses the many that are ideally suited for role playing, strategy or adventure gaming, the actual rate of automated strategy and randomly generated events is incredibly disappointing.

The game begins quite well with an impressive theme of the sort of music and graphics even a C64, PC is capable of, but if you get into the game, it's all downhill from here. You begin the game with a choice of four explorers and an expert guide, although after a rock fall you're on your own. The rockfall is the first of the "events", but none of this.

Rapidly, the game drops into its headram routine and upon a screen showing the full and incredibly limited map that

you have to navigate, displays showing your food, water, vitality and physical condition. There are also a score of icons, which must be selected to move, rest (to conserve energy), ration (to spread your food further) and one that leads to the mode screen. Here you can get a diagnosis of your life, and there's a first aid kit to patch them up.

Whenever you click the mouse icon you hear a few quiet footprints and something may happen, ranging from a silly event to an even sillier outside sequence. For example events include snakes that don't die, but you can slay; pebbles that trip you; and bumps on the head that turn into fractured skulls just because of a badly placed statistic. As for the arcade sequences, try these two for examples. In the repetitive falling rock game you have to run up three steps while avoiding the rocks. Yes, you read this correctly, you run up even though you're going down into the Earth!

The second example is when you have to avoid a herd of stamping elephants! Exactly what a herd of elephants is doing beneath the earth's surface isn't explained, neither is how the thin tunnel you were in suddenly becomes large enough for a congregation of mammoths.

I find this the most annoying type of game as it promises so much, looks and sounds so good and delivers so little.

Tony Hetherington

INFO

Gameplay: 19%
Graphics: 67%
Sound: 63%
Lastability: 20%
Overall:
42%

Journey to the Centre of the Earth is produced by French developers Chip, who also wrote Jour de l'été marketed by US Gold.



Extending Basic

Get a printout of the variables of your Basic program with this latest routine in our series on extended Basic.

By Bernhard-Henry Lehmann

In this series of articles on developing extended Basic routines, we have already dealt, several times, with the way Basic stores its variables. Therefore, the next logical step is to write a routine which gives us a printout of the variables created by a Basic program and what they contain at any given time.

This is a very easy thing to do, because all that's required is that we loop through the Basic variable areas and print each variable name and content.

I wanted to make this facility possible as a direct command and in connection with the trace facility which we developed in the last article. I am including this in this installment for all you people who didn't buy last month's **YOUR COMMODORE** (same on post).

As a direct command, you simply

enter **VAR**, and all variables of the Basic program which you have just run are printed out. Of course, if you haven't run a Basic program or have given the direct **CLR** command, which clears all Basic variables, nothing will be printed out. The same is true if you have changed the Basic terfite. As you know, the moment you enter a Basic line by pressing return all variables are discarded.

In connection with the trace facility which we developed last month, you can call a printout of all variables after each line has been executed and the trace routine has replaced the line at the top of the screen. You will remember, that after each tracing stop, the computer waits until you press any key. To get a printout of the variables simply press "V". If you want program execution to continue press any other key.

How Basic Stores Its Variables

Before we look closer at the program itself, let's briefly recapitulate how Basic stores its variables.

All Basic variables are stored directly after the end of the Basic terfite. To find the start of the variable area you have to push the auto page variable decimal 43/45, Hex 2D/2E. Each variable takes up seven bytes of memory.

The first two bytes contain the variable name. The way the variable name is entered shows what type of variable we are dealing with:

A floating point variable has both bytes of its name entered in simple ASCII. If the name consists of only one letter, the second byte contains zero.

An integer variable has 128 added

in both its name characters, and if there is only one character (which, as you remember, always has to be a letter), the second byte contains 128. (All this means, that in such case, the seven of the eight binary bits is set.)

A string variable is distinguished by having the letter of its name entered in simple ASCII, while the second character has 128 added to the ASCII value, or 128, if there isn't a second character.

Finally, a function definition has 128 added to the first letter of its name, while the second character is simple ASCII or zero. (Incidentally, in the program given I have ignored function definitions. As always, this is partly due to my laziness and partly in order to encourage you to add this to the routine yourself...)

...The final five bytes of each variable contain the contents of the variable itself. This is correct in the case of the string variables. Here, the third byte contains the length of the string and the fourth and the fifth byte contain the base address of the string in the Basic textfile.

This is quite a clever way of going about things. Instead of storing the string itself in the variable area, which would of course take up codan of memory and make the whole variable area pretty difficult to manage, a pointer address which takes up only two bytes is stored in the variable area. The disadvantage with this method is of course, that the Basic textfile must not be changed. The moment it is changed, the pointer to a certain string in the variable area might not be correct any more. This is one of the reasons why you can't continue running a Basic program after you have changed the textfile.

The storage of an integer variable is also pretty straightforward. The value of the variable is stored in the third and fourth byte after the name, whereby the third byte contains the high byte of the value, plus the so called sign bit, and fourth byte contains the low byte. You assign an integer variable by entering "N" after it, e.g. 8%.

An integer ranges from -32768 to +32767. If you add these two numbers up you get 65535, which is 65536 short of one, because the zero is included as a valid number. And 65536 is the

maximum range of a 16-bit address. If bit seven, or the sign bit, of the high byte is set, the number is considered to be negative, if it is clear, the number is considered to be positive. For example, high 0 and low 160 represent +160, while high 255 and low 155 represent -100.

The most complicated variables of all are the floating point variables. This is a pretty long subject which I do not want to enter at this point. There is also no need to, because in our program we are using ROM-routines to convert the floating point number as it is stored in the variable area into an ASCII string which we then simply print out.

But, for the mathematicians amongst you, here is how a floating point number is stored in the Basic variable area: The byte after the variable name contains the exponent, the byte after that contains the first mantissa, plus the sign bit, and the final three bytes contain mantissa two to four.

Printing the Variables

The routine which prints each variable name and the contents of the variable, is so straightforward, that it needs very little explanation.

The main loop, which I call VARS-LOOP, first of all prints the variable name and decides at the same time what type of variable has to be dealt with (lines 4180-4210). According to those tests the program flow then jumps to the appropriate routine: FLPOINT, if it is a floating point variable, INTEGER, if it is an integer variable, and STRING, if it is a string variable.

Before the value of each type of variable is printed, some spaces and an equal sign are printed, as well as the "N"-sign, in case of a string, and the "I"-sign, in case of an integer. This makes the printed more presentable.

In the case of a floating point variable, A is pointed at the low byte of the first byte of the floating point value and Y is pointed at the high byte (lines 4470-4540).

Then the ROM-routine at \$BBA2 is called to put the value into the first of the two floating point accumulators.

Next, the ROM-routine at \$BDDDD converts the contents of the

floating point accumulator into a printable ASCII string. On exit from that routine, A and Y point to the base of that string.

Finally, the string is printed in the usual fashion, with \$ABEE, which prints any string point at by A/Y and terminated with a zero.

An integer variable is printed by loading its high byte into A and its low byte into X. The ROM-routine at \$BDCD, which prints any number contained in A/X, prints the integer value out.

To print a string variable, we first get the length of the string and store it in Z (lines 5450-5470). Then we transfer the address which points at the string in the textfile into zero page 251/254 (lines 5510-5560).

Finally, we use indexes-Y to print the string, letter by letter. This is done with a loop which is limited by the length value contained in Z (lines 5680-5690).

After a variable has been dealt with, VARSLOOP ends by incrementing the pointer address by seven (lines 4770-4830). The vital loop exit test, which tells when there are no more variables, is done at the beginning of VARSLOOP (lines 4070-4120). This is because the routine has to exit at once if there are no variables at all.

The end of the Basic variables area is contained in zero page 32F/30. If this point is reached, there are no more variables. What follows are the Basic arrays with which we are not dealing at this point.

Final Notes

As I was developing the variables printed routine I discovered to my great puzzlement that certain variable names did not work properly. For example "CC" was stored as "C" and "CT" didn't work at all and resulted in a syntax error report.

Then I discovered the reason for these irregularities: The extended Basic commands we use starts with a letter, that is a normal ASCII character. If you declare a variable by entering the LET command (as one usually does), then the variable name too starts with a letter!

To overcome this I changed the main routine in the following way: First, I added two lines, which use for

a status character (lines 550-560). If a token has been found, which means that the line starts with a Basic command, the routine jumps straight to \$ATED which evaluates the Basic command in the usual manner.

Next, before starting the tests for our extended Basic commands, I save the Basic pointer \$PA/78 (lines 600-610). If the tests are negative, that is, if none of our extended Basic commands has been entered, \$PA/78 is recovered (lines 1070-1080). This resets the pointer to the first character on the line, resulting in a variable name being read correctly.



```

100 10000
110 10100
120 10200
130 10300
140 10400
150 10500
160 10600
170 10700
180 10800
190 10900
200 11000
210 11100
220 11200
230 11300
240 11400
250 11500
260 11600
270 11700
280 11800
290 11900
300 12000
310 12100
320 12200
330 12300
340 12400
350 12500
360 12600
370 12700
380 12800
390 12900
400 13000
410 13100
420 13200
430 13300
440 13400
450 13500
460 13600
470 13700
480 13800
490 13900
500 14000
510 14100
520 14200
530 14300
540 14400
550 14500
560 14600
570 14700
580 14800
590 14900
600 15000
610 15100
620 15200
630 15300
640 15400
650 15500
660 15600
670 15700
680 15800
690 15900
700 16000
710 16100
720 16200
730 16300
740 16400
750 16500
760 16600
770 16700
780 16800
790 16900
800 17000
810 17100
820 17200
830 17300
840 17400
850 17500
860 17600
870 17700
880 17800
890 17900
900 18000
910 18100
920 18200
930 18300
940 18400
950 18500
960 18600
970 18700
980 18800
990 18900
1000 19000
1010 19100
1020 19200
1030 19300
1040 19400
1050 19500
1060 19600
1070 19700
1080 19800
1090 19900
1100 20000
1110 20100
1120 20200
1130 20300
1140 20400
1150 20500
1160 20600
1170 20700
1180 20800
1190 20900
1200 21000
1210 21100
1220 21200
1230 21300
1240 21400
1250 21500
1260 21600
1270 21700
1280 21800
1290 21900
1300 22000
1310 22100
1320 22200
1330 22300
1340 22400
1350 22500
1360 22600
1370 22700
1380 22800
1390 22900
1400 23000
1410 23100
1420 23200
1430 23300
1440 23400
1450 23500
1460 23600
1470 23700
1480 23800
1490 23900
1500 24000
1510 24100
1520 24200
1530 24300
1540 24400
1550 24500
1560 24600
1570 24700
1580 24800
1590 24900
1600 25000
1610 25100
1620 25200
1630 25300
1640 25400
1650 25500
1660 25600
1670 25700
1680 25800
1690 25900
1700 26000
1710 26100
1720 26200
1730 26300
1740 26400
1750 26500
1760 26600
1770 26700
1780 26800
1790 26900
1800 27000
1810 27100
1820 27200
1830 27300
1840 27400
1850 27500
1860 27600
1870 27700
1880 27800
1890 27900
1900 28000
1910 28100
1920 28200
1930 28300
1940 28400
1950 28500
1960 28600
1970 28700
1980 28800
1990 28900
2000 29000
2010 29100
2020 29200
2030 29300
2040 29400
2050 29500
2060 29600
2070 29700
2080 29800
2090 29900
2100 30000
2110 30100
2120 30200
2130 30300
2140 30400
2150 30500
2160 30600
2170 30700
2180 30800
2190 30900
2200 31000
2210 31100
2220 31200
2230 31300
2240 31400
2250 31500
2260 31600
2270 31700
2280 31800
2290 31900
2300 32000
2310 32100
2320 32200
2330 32300
2340 32400
2350 32500
2360 32600
2370 32700
2380 32800
2390 32900
2400 33000
2410 33100
2420 33200
2430 33300
2440 33400
2450 33500
2460 33600
2470 33700
2480 33800
2490 33900
2500 34000
2510 34100
2520 34200
2530 34300
2540 34400
2550 34500
2560 34600
2570 34700
2580 34800
2590 34900
2600 35000
2610 35100
2620 35200
2630 35300
2640 35400
2650 35500
2660 35600
2670 35700
2680 35800
2690 35900
2700 36000
2710 36100
2720 36200
2730 36300
2740 36400
2750 36500
2760 36600
2770 36700
2780 36800
2790 36900
2800 37000
2810 37100
2820 37200
2830 37300
2840 37400
2850 37500
2860 37600
2870 37700
2880 37800
2890 37900
2900 38000
2910 38100
2920 38200
2930 38300
2940 38400
2950 38500
2960 38600
2970 38700
2980 38800
2990 38900
3000 39000
3010 39100
3020 39200
3030 39300
3040 39400
3050 39500
3060 39600
3070 39700
3080 39800
3090 39900
3100 40000
3110 40100
3120 40200
3130 40300
3140 40400
3150 40500
3160 40600
3170 40700
3180 40800
3190 40900
3200 41000
3210 41100
3220 41200
3230 41300
3240 41400
3250 41500
3260 41600
3270 41700
3280 41800
3290 41900
3300 42000
3310 42100
3320 42200
3330 42300
3340 42400
3350 42500
3360 42600
3370 42700
3380 42800
3390 42900
3400 43000
3410 43100
3420 43200
3430 43300
3440 43400
3450 43500
3460 43600
3470 43700
3480 43800
3490 43900
3500 44000
3510 44100
3520 44200
3530 44300
3540 44400
3550 44500
3560 44600
3570 44700
3580 44800
3590 44900
3600 45000
3610 45100
3620 45200
3630 45300
3640 45400
3650 45500
3660 45600
3670 45700
3680 45800
3690 45900
3700 46000
3710 46100
3720 46200
3730 46300
3740 46400
3750 46500
3760 46600
3770 46700
3780 46800
3790 46900
3800 47000
3810 47100
3820 47200
3830 47300
3840 47400
3850 47500
3860 47600
3870 47700
3880 47800
3890 47900
3900 48000
3910 48100
3920 48200
3930 48300
3940 48400
3950 48500
3960 48600
3970 48700
3980 48800
3990 48900
4000 49000
4010 49100
4020 49200
4030 49300
4040 49400
4050 49500
4060 49600
4070 49700
4080 49800
4090 49900
4100 50000
4110 50100
4120 50200
4130 50300
4140 50400
4150 50500
4160 50600
4170 50700
4180 50800
4190 50900
4200 51000
4210 51100
4220 51200
4230 51300
4240 51400
4250 51500
4260 51600
4270 51700
4280 51800
4290 51900
4300 52000
4310 52100
4320 52200
4330 52300
4340 52400
4350 52500
4360 52600
4370 52700
4380 52800
4390 52900
4400 53000
4410 53100
4420 53200
4430 53300
4440 53400
4450 53500
4460 53600
4470 53700
4480 53800
4490 53900
4500 54000
4510 54100
4520 54200
4530 54300
4540 54400
4550 54500
4560 54600
4570 54700
4580 54800
4590 54900
4600 55000
4610 55100
4620 55200
4630 55300
4640 55400
4650 55500
4660 55600
4670 55700
4680 55800
4690 55900
4700 56000
4710 56100
4720 56200
4730 56300
4740 56400
4750 56500
4760 56600
4770 56700
4780 56800
4790 56900
4800 57000
4810 57100
4820 57200
4830 57300
4840 57400
4850 57500
4860 57600
4870 57700
4880 57800
4890 57900
4900 58000
4910 58100
4920 58200
4930 58300
4940 58400
4950 58500
4960 58600
4970 58700
4980 58800
4990 58900
5000 59000
5010 59100
5020 59200
5030 59300
5040 59400
5050 59500
5060 59600
5070 59700
5080 59800
5090 59900
5100 60000
5110 60100
5120 60200
5130 60300
5140 60400
5150 60500
5160 60600
5170 60700
5180 60800
5190 60900
5200 61000
5210 61100
5220 61200
5230 61300
5240 61400
5250 61500
5260 61600
5270 61700
5280 61800
5290 61900
5300 62000
5310 62100
5320 62200
5330 62300
5340 62400
5350 62500
5360 62600
5370 62700
5380 62800
5390 62900
5400 63000
5410 63100
5420 63200
5430 63300
5440 63400
5450 63500
5460 63600
5470 63700
5480 63800
5490 63900
5500 64000
5510 64100
5520 64200
5530 64300
5540 64400
5550 64500
5560 64600
5570 64700
5580 64800
5590 64900
5600 65000
5610 65100
5620 65200
5630 65300
5640 65400
5650 65500
5660 65600
5670 65700
5680 65800
5690 65900
5700 66000
5710 66100
5720 66200
5730 66300
5740 66400
5750 66500
5760 66600
5770 66700
5780 66800
5790 66900
5800 67000
5810 67100
5820 67200
5830 67300
5840 67400
5850 67500
5860 67600
5870 67700
5880 67800
5890 67900
5900 68000
5910 68100
5920 68200
5930 68300
5940 68400
5950 68500
5960 68600
5970 68700
5980 68800
5990 68900
6000 69000
6010 69100
6020 69200
6030 69300
6040 69400
6050 69500
6060 69600
6070 69700
6080 69800
6090 69900
6100 70000
6110 70100
6120 70200
6130 70300
6140 70400
6150 70500
6160 70600
6170 70700
6180 70800
6190 70900
6200 71000
6210 71100
6220 71200
6230 71300
6240 71400
6250 71500
6260 71600
6270 71700
6280 71800
6290 71900
6300 72000
6310 72100
6320 72200
6330 72300
6340 72400
6350 72500
6360 72600
6370 72700
6380 72800
6390 72900
6400 73000
6410 73100
6420 73200
6430 73300
6440 73400
6450 73500
6460 73600
6470 73700
6480 73800
6490 73900
6500 74000
6510 74100
6520 74200
6530 74300
6540 74400
6550 74500
6560 74600
6570 74700
6580 74800
6590 74900
6600 75000
6610 75100
6620 75200
6630 75300
6640 75400
6650 75500
6660 75600
6670 75700
6680 75800
6690 75900
6700 76000
6710 76100
6720 76200
6730 76300
6740 76400
6750 76500
6760 76600
6770 76700
6780 76800
6790 76900
6800 77000
6810 77100
6820 77200
6830 77300
6840 77400
6850 77500
6860 77600
6870 77700
6880 77800
6890 77900
6900 78000
6910 78100
6920 78200
6930 78300
6940 78400
6950 78500
6960 78600
6970 78700
6980 78800
6990 78900
7000 79000
7010 79100
7020 79200
7030 79300
7040 79400
7050 79500
7060 79600
7070 79700
7080 79800
7090 79900
7100 80000
7110 80100
7120 80200
7130 80300
7140 80400
7150 80500
7160 80600
7170 80700
7180 80800
7190 80900
7200 81000
7210 81100
7220 81200
7230 81300
7240 81400
7250 81500
7260 81600
7270 81700
7280 81800
7290 81900
7300 82000
7310 82100
7320 82200
7330 82300
7340 82400
7350 82500
7360 82600
7370 82700
7380 82800
7390 82900
7400 83000
7410 83100
7420 83200
7430 83300
7440 83400
7450 83500
7460 83600
7470 83700
7480 83800
7490 83900
7500 84000
7510 84100
7520 84200
7530 84300
7540 84400
7550 84500
7560 84600
7570 84700
7580 84800
7590 84900
7600 85000
7610 85100
7620 85200
7630 85300
7640 85400
7650 85500
7660 85600
7670 85700
7680 85800
7690 85900
7700 86000
7710 86100
7720 86200
7730 86300
7740 86400
7750 86500
7760 86600
7770 86700
7780 86800
7790 86900
7800 87000
7810 87100
7820 87200
7830 87300
7840 87400
7850 87500
7860 87600
7870 87700
7880 87800
7890 87900
7900 88000
7910 88100
7920 88200
7930 88300
7940 88400
7950 88500
7960 88600
7970 88700
7980 88800
7990 88900
8000 89000
8010 89100
8020 89200
8030 89300
8040 89400
8050 89500
8060 89600
8070 89700
8080 89800
8090 89900
8100 90000
8110 90100
8120 90200
8130 90300
8140 90400
8150 90500
8160 90600
8170 90700
8180 90800
8190 90900
8200 91000
8210 91100
8220 91200
8230 91300
8240 91400
8250 91500
8260 91600
8270 91700
8280 91800
8290 91900
8300 92000
8310 92100
8320 92200
8330 92300
8340 92400
8350 92500
8360 92600
8370 92700
8380 92800
8390 92900
8400 93000
8410 93100
8420 93200
8430 93300
8440 93400
8450 93500
8460 93600
8470 93700
8480 93800
8490 93900
8500 94000
8510 94100
8520 94200
8530 94300
8540 94400
8550 94500
8560 94600
8570 94700
8580 94800
8590 94900
8600 95000
8610 95100
8620 95200
8630 95300
8640 95400
8650 95500
8660 95600
8670 95700
8680 95800
8690 95900
8700 96000
8710 96100
8720 96200
8730 96300
8740 96400
8750 96500
8760 96600
8770 96700
8780 96800
8790 96900
8800 97000
8810 97100
8820 97200
8830 97300
8840 97400
8850 97500
8860 97600
8870 97700
8880 97800
8890 97900
8900 98000
8910 98100
8920 98200
8930 98300
8940 98400
8950 98500
8960 98600
8970 98700
8980 98800
8990 98900
9000 99000
9010 99100
9020 99200
9030 99300
9040 99400
9050 99500
9060 99600
9070 99700
9080 99800
9090 99900
9100 100000
9110 100100
9120 100200
9130 100300
9140 100400
9150 100500
9160 100600
9170 100700
9180 100800
9190 100900
9200 101000
9210 101100
9220 101200
9230 101300
9240 101400
9250 101500
9260 101600
9270 101700
9280 101800
9290 101900
9300 102000
9310 102100
9320 102200
9330 102300
9340 102400
9350 102500
9360 102600
9370 102700
9380 102800
9390 102900
9400 103000
9410 103100
9420 103200
9430 103300
9440 103400
9450 103500
9460 103600
9470 103700
9480 103800
9490 103900
9500 104000
9510 104100
9520 104200
9530 104300
9540 104400
9550 104500
9560 104600
9570 104700
9580 104800
9590 104900
9600 105000
9610 105100
9620 105200
9630 105300
9640 105400
9650 105500
9660 105600
9670 105700
9680 105800
9690 105900
9700 106000
9710 106100
9720 106200
9730 106300
9740 106400
9750 106500
9760 106600
9770 106700
9780 106800
9790 106900
9800 107000
9810 107100
9820 107200
9830 107300
9840 107400
9850 107500
9860 107600
9870 107700
9880 107800
9890 107900
9900 108000
9910 108100
9920 108200
9930 108300
9940 108400
9950 108500
9960 108600
9970 108700
9980 108800
9990 108900
10000 109000
10010 109100
10020 109200
10030 109300
10040 109400
10050 109500
10060 109600
10070 109700
10080 109800
10090 109900
10100 110000
10110 110100
10120 110200
10130 110300
10140 110400
10150 110500
10160 110600
10170 110700
10180 110800
10190 110900
10200 111000
10210 111100
10220 111200
10230 111300
10240 111400
10250 111500
10260 111600
10270 111700
10280 111800
10290 111900
10300 112000
10310 112100
10320 112200
10330 112300
10340 112400
10350 112500
10360 112600
10370 112700
10380 112800
10390 112900
10400 113000
10410 113100
10420 113200
10430 113300
10440 113400
10450 113500
10460 113600
10470 113700
10480 113800
10490 113900
10500 114000
10510 114100
10520 114200
10530 114300
10540 114400
10550 114500
10560 114600
10570 114700
10580 114800
10590 114900
10600 115000
10610 115100
10620 115200
10630 115300
10640 115400
10650 115500
10660 115600
10670 115700
10680 115800
10690 115900
10700 116000
10710 116100
10720 116200
10730 116300
10740 116400
10750 116500
10760 116600
10770 116700
10780 116800
10790 116900
10800 117000
10810 117100
10820 117200
10830 117300
10840 117400
10850 117500
10860 117600
10870 1177
```




giving you with us just as, like meeting the rest of the magazine. BREAK is to have an as comprehensive new era. This is now the magazine where you'll be able to read all the policy groups that is floating around the Commodore market.

The Lines are Blurred.

A Reading resident has been having a rather harassing time of late due to an *Amesbury* magazine who misprinted the Activision hotline number. The person in question was receiving calls all around the clock from people requesting hints and tips on Activision games - causing some distress. The correct number is 0177349 118003.

Speaking of telephone lines, those kind folks at British Telecom have changed Incentive's number. The new number and code is, if you're at all interested, (02346) 617288.

Add: Sir Joz

As Ford, the editor of *Electronics Today International*—one of our many sister magazines. Now the Arms Pa-

On the recent Lone Wolf press launch, when he came away with the going in a rather impressive. Out of the thirty-plus press representatives, he was the first to solve many clues and discover the identity of Lone Wolf (a teenage thief in costume).



"If [Lester] it wasn't for a certain Deputy [Lester's] house cleaning, the result may have turned out very different."

Just When You

...Thought you could get your hands on *Mass Cops*, the game has been delayed yet again.

Due originally for release almost a year ago, the game was nearing completion when the Bubble Sugar of late struck Acclaim Software Limited. The programmer, who had worked to getting the game two weeks from being ready, has walked out on the company. This means a new programmer will have to finish the project and thus we shall not see Mars Cops until the beginning of November.

Commodore Where Are You?

As we constantly get asked for the telephone number of Commodore U.K., it is best if we just print it here:
Tel: 0452 330000

The Middle East After 9/11



Emlyn Hughes



INTERNATIONAL

SOCCER



**GAME
OF THE YEAR!**

**WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!**

Quite simply the best football
simulation available for the 64.
Nothing short of superb.

ZZAP 64

**AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk**

Coming soon for Spectrum & Amstrad

Audiogenic

Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SL, England

Order by phone on 01 861 1166

